ROBOTS LEARNING ACTIONS AND GOALS FROM EVERYDAY PEOPLE

A Thesis
Presented to
The Academic Faculty

by

Baris Akgun

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology
December 2015

ROBOTS LEARNING ACTIONS AND GOALS FROM EVERYDAY PEOPLE

Approved by:

Professor Andrea L. Thomaz, Adviser
School of Interactive Computing
Georgia Institute of Technology

Professor Magnus Egerstedt
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Henrik Christensen
School of Interactive Computing
Georgia Institute of Technology

Professor Pieter Abbeel
School of Electrical Engineering and Computer Sciences
University of California Berkeley

Professor Charles Isbell
School of Interactive Computing
Georgia Institute of Technology

Date Approved: 9 November 2015

ACKNOWLEDGEMENTS

I am privileged to have had Andrea Thomaz as my PhD advisor. She emanates an aura of "professorship". I knew from our first meeting that if she took me in as a student, I would be in good hands. She genuinely cared about me, my research and my future. She gave the right amount of guidance, advise and push at the right moments. She also set an example of how you can achieve good work-life balance. She will be a role model throughout my career.

I am indebted to my thesis committee members, Henrik Christensen, Pieter Abbeel, Magnus Egerstedt, and Charles Isbell for their time and patience. Their input and feedback helped me to raise the quality of my work. I want to thank Henrik especially for leading the robotics charge at Georgia Tech and providing us with this amazing environment. I had the opportunity of working with the late Mike Stilman, who had the most passion for robots I have ever seen. I also want to thank Charlie Kemp for highly informative discussions about my research early on in my thesis.

I was lucky to be part of the Socially Intelligent Machines Lab during my thesis. I am grateful to my academic sister Maya Cakmak for her mentorship during the first two years of my PhD. She helped me hit the ground running when I joined the lab. She also set an example for the rest of us to follow. I have enjoyed our deep conversations about the "big questions" such as "What is the point?" and "Is this worth it?" with my lab mate Crystal Chao. It was a pleasure to share the same cubicle with such a talented researcher and programmer. Right when everybody was either graduating or going somewhere, Vivian Chu, Kalesha Bullard and Tesca Fitzgerald joined the lab and brought a breath of fresh air. They made me feel "senior" in a good way. Vivian inspired me with her never ending optimism and love for organization. Kalesha was always there whenever I needed to procrastinate.

I would also like to thank other past and current members of the lab, Andrey Kurenkov, Shane Griffith, Jaewook Joo, Yannick Schroecker and Eric Huang.

I am thankful to have found many close friends in my time at Georgia Tech. From coffee breaks, to technical discussions to road trips, Martin Levihn and Jon Scholz were always there whenever needed. I could always count on them for advice. It was always a joy to spend time with Martin with his love of gaming, sense of humor, unrelenting personality and propensity for dropping hard facts along with even harder opinions. Jon, with his interdisiplinary vision, always saw the both sides of the coin. He made me realize the light at the end of the tunnel is real whenever I had doubts. He was always fun to hang out with, he always had an interesting plan, from road trips to parties. I think we would all have graduated sooner but it wouldn't be half as much fun and memorable. I am glad Heni Ben Amor, an old acquaintance, joined Georgia Tech for a post-doc. He provided a unique perspective on everything in academia and had a wide breadth of knowledge in robotics. I want to thank Ahmet Ceyhan for the all the conversations from sports to politics to science. Akansel Cosgun was always available to chat or rant about stuff in my native language. Misha Novitzky was one of the nicest people I've met in Atlanta. He was always positive, cheerful, and eager to help. When all of us was getting boring and tired Sasha Lambert joined the program and gave us new energy. Paul Robinette was the perfect study partner for the qualifiers and he was always kind enough to invite me at their home for amazing cooking. Kaushik Subramanian's perspective on research made me think differently at certain point of my thesis. Pushkar Kolhe has always had a unique life style and views which made conversations with him highly enjoyable and interesting. I am glad that Himanshu Sahni joined our floor and help make my last year in graduate school more fun. I would like to further thank Alex Trevor, Rowland O'Flaherty, Rahul Sawhney, Brian Goldfain, Brian Hrolenok, Changhyun Choi, and Can Erdogan for their friendship throughout my PhD.

I appreciate all the support of my friends from Turkey who are too numerous to name

here. I am sorry that I was not available enough. I knew you were expecting great things from me and I will do my best to live up to the expectation.

I am lucky to have had access to amazing robots which made my thesis better. I would like to thank Meka Robotics, the makers of the robots Simon and Curi, and Willow Garage, the makers of the robot PR2. It is a shame that both of these companies do not exist anymore.

I want to thank all my family members for supporting and enduring me. I can never repay my mom and dad for all the things they do for me. I am grateful for their infinite support, even if I don't always show it. Hearing my sister's joyful voice, "how are you my dear little brother!", has always been energizing. I want to thank my grandmothers for all the encouragement they gave me. I wish both of my grandfathers were alive to see me graduate. I would like to thank my cousins, aunts and uncles for bearing with me even when I was really bad at keeping touch.

Finally, I want to express my gratitudes to the love of my life, Nazlı, for putting up with me, supporting me and for always being there. She made even the worst days feel better with her smile. In addition to supporting me, she inspired me as a fellow academic. Everything is more meaningful because of her and I am glad to have had her by my side.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Robots are destined to move beyond the caged factory floors towards domains where they will be interacting closely with humans. They will encounter highly varied environments, scenarios and user demands. As a result, programming robots after deployment will be an important requirement. To address this challenge, the field of Learning from Demonstration (LfD) emerged with the vision of programming robots through demonstrations of the desired behavior instead of explicit programming. The field of LfD within robotics has been around for more than 30 years and is still an actively researched field. However, very little research is done on the implications of having a non-robotics expert as a teacher. This thesis aims to bridge this gap by developing learning from demonstration algorithms and interaction paradigms that allow non-expert people to teach robots new skills.

The first step of the thesis was to evaluate how non-expert teachers provide demonstrations to robots. Keyframe demonstrations are introduced to the field of LfD to help people teach skills to robots and compared with the traditional trajectory demonstrations. The utility of keyframes are validated by a series of experiments with more than 80 participants. Based on the experiments, a hybrid of trajectory and keyframe demonstrations are proposed to take advantage of both and a method was developed to learn from trajectories, keyframes and hybrid demonstrations in a unified way.

A key insight from these user experiments was that teachers are goal oriented. They concentrated on achieving the goal of the demonstrated skills rather than providing good quality demonstrations. Based on this observation, this thesis introduces a method that can learn actions and goals from the same set of demonstrations. The action models are used to execute the skill and goal models to monitor this execution. A user study with eight

participants and two skills showed that successful goal models can be learned from non-expert teacher data even if the resulting action models are not as successful. Following these results, this thesis further develops a self-improvement algorithm that uses the goal monitoring output to improve the action models, without further user input. This approach is validated with an expert user and two skills. Finally, this thesis builds an interactive LfD system that incorporates both goal learning and self-improvement and evaluates it with 12 naive users and three skills. The results suggests that teacher feedback during experiments increases skill execution and monitoring success. Moreover, non-expert data can be used as a seed to self-improvement to fix unsuccessful action models.

CHAPTER I

INTRODUCTION

This thesis aims at developing learning from demonstration algorithms and interaction paradigms that allow non-expert people to teach robots new skills. Robots are getting cheaper and more accessible. As they are becoming more ubiquitous, the range of tasks and environments they face are growing exponentially more complex. Many of these environments, such as households, hospitals, and schools, contain people having a wide range of preferences, expectations, assumptions, and level of technological savviness. Although many of these people know what they want the robot to do, almost all of them have little or no knowledge about the robot itself. The central goal of the research in this thesis is to close this gap between what the humans think and how they communicate, and the level at which robot algorithms operate.

It is difficult to program robots for the scenarios that they will face when they are deployed for such application domains. Moreover, there will be cases where end-users of these robots who are not satisfied with the existing programs will want to customize programs for their own preferences. As a result, programming of robots after their deployment becomes a necessity. An idea is to let the end-users program their own skills. However, programming a robot is not easy, especially for a person without a background in robotics and computer science. In order to address this challenge, the field of *Learning from Demonstration* (LfD) [24] emerged with the vision of programming robots through demonstrations of the desired behavior instead of explicit programming. An instance of a human using LfD to teach a robot a skill can be seen in Fig. 1.

An overview of a typical LfD system can be seen in Fig. 2. A teacher demonstrates the desired skill to a robot via an input modality, *e.g.* with guiding robot's arm as in Fig. 1. The

Figure 1: Learning from Everyday People: A non-expert teaching a robot how to pour. In this interaction, the teacher guides the robot's arm to demonstrate the skill.

robot records data such as joint or end-effector poses. This data is then input to a learning algorithm which yields a skill model. This model can be executed on the robot. In some methods, the robot improves this model based on a pre-defined reward function as depicted by the loop between the robot and the algorithm. The LfD process can also be interactive in which the teacher observes the robot execute the skill.

There is a significant amount of existing work in the field of LfD (see Chapter 2). Most existing work has concentrated on the *Algorithm* block of Fig. 2, ignoring implications of the interaction with the teacher. Hence, there are not many user studies to evaluate these methods to validate the assumptions they make about data and end-users. End-users may not always be able to provide the data required for these algorithms to work as intended, for example, due to not being able to operate the robot well or having incorrect assumptions about how the robot learns. They may also not have the time to provide the wide range of demonstrations needed to learn models that generalize to a variety of contexts. By and

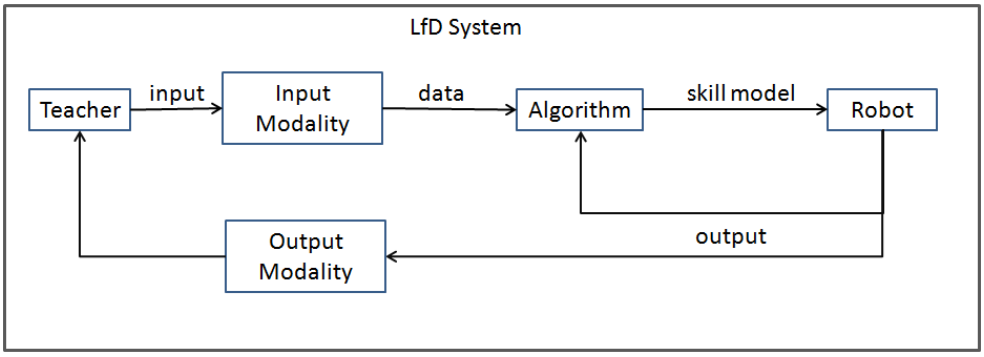


Figure 2: An example LfD system depicting the process as a loop. This diagram captures most of the existing work on skill learning, which are usually subsets of the system.

large, the applicability of the existing methods by a variety of human teachers was not addressed in the literature.

This goal of this thesis research is to enhance robotic learning from demonstration for naïve human teachers, *i.e.* people who do not have robotics or machine learning expertise, such that they can teach a wide variety of skills to a robot or customize the already available skills. There are several challenges arising from this scenario:

1. The demonstrations should be easy for a teacher to provide.

2. The system needs to learn efficiently from few interactions with the teacher.

3. The system needs to learn a variety of skills without tuning any aspects of the methods for each skill.

4. The demonstrations of the teachers may be noisy and inconsistent but still capture the essence of the skill or intent of the user.

This thesis takes a Human Robot Interaction (HRI) approach to LfD to tackle these challenges. The described research addresses both the interactions and the algorithms that are best suited for LfD to learn a wide range of skills, rather than just focusing on the algorithms block of Fig. 2. Therefore, this thesis takes a holistic approach to LfD by looking at a range of topics from teacher demonstrations to robot executions. The entire the LfD

process is regarded as an interactive loop between the teacher and the robot, as depicted in Fig. 2.

1.1 Thesis Statement

A human centered, interactive, goal oriented and self-improving approach to learning from demonstration that learns both action and goal models, increases skill performance compared to only learning action models.Human centered refers to both the interaction side of learning from demonstration, in the form of involving non-expert teachers and the representation side, in the form of keyframes.

1.2 Summary of Contributions

This thesis introduces and evaluates algorithms to address these interactive LfD challenges. The following contributions are made in this thesis:

- **Keyframes**: Keyframe demonstrations are introduced to the field of LfD to help people teach skills to robots. An experiment, *Experiment I* with 34 users compared keyframe demonstrations versus trajectory demonstrations [4] and showed that both have their respective advantages. A hybrid of trajectory and keyframe demonstrations are proposed to take advantage of both. A method, *Keyframe based Learning from Demonstration - KLfD*, is introduced to learn from trajectory, keyframe and hybrid demonstrations in a unified way [3]. Another experiment, *Experiment II*, evaluated trajectories, keyframes and hybrid demonstrations for teleoperation in the context of LfD with 21 participants [6]. These evaluations showed that both keyframe representation and hybrid demonstrations are valuable tools to enable non-expert people teach skills to robots.

- **Goal Learning**: Experiments I and II showed that people are goal oriented and this is reflected the way they teach robots. Majority of the users did not mind providing

noisy, inconsistent demonstrations with unnecessary motions as long as they successfully demonstrated the skill. This insight inspired a new way of thinking about LfD and led to the development of *Goal and Action Learning from Demonstration - GoaLfD* framework, to learn both action and goal models from the same set of user demonstrations. After learning, action models are used to execute the skill and the goal models are used to monitor this execution. GoaLfD was tested with a pilot study involving 8 participants [7] and an experiment, *Experiment III*, with another 8 participants [9]. These showed that goal models are able to correctly monitor the skill executions even if the accompanying action models are not as successful. Robots can use this capability to know whether their executions succeeded.

- **Self-Improvement**: The Experiment III showed that non-expert people may not be able to teach successful action models. In addition, these teachers have limited time and patience. They may not be able to fix their action models or demonstrate the entire envelope of the skill. This thesis introduces a self-improvement algorithm based on the GoaLfD framework, *Goal based Learning and Exploration - GoaL-E* that updates the action models to both fix them and to increase their execution envelope based on goal model output [8]. The algorithm samples from the learned action model to execute on the robot and uses the monitoring result as the self-improvement signal. An adaptive sampling method is introduced to trade-off exploration and exploitation during self-improvement. The results show that the algorithm can fix a failing action model using a successful goal model

- **Interactive Learning**: In human-human teaching, parties interact to communicate each other's states to come to common ground and facilitate efficient learning. Similarly, knowing the robot's learning state would allow the teacher to tailor his/her demonstrations accordingly and increase learning efficiency. However, this is harder than the human-human case since the way a robot learns and communicates is very

different than a human. The first step towards this end is the implicit communication of the robot's learning state that happens when the teacher sees the robot executing the learned skill during the learning process. The Experiments I and II had an interactive component but this was not the main point in those experiments. The Experiment III concentrated on testing the action and goal learning with non-expert teachers and did not have an interactive component. As the final contribution, this thesis developed an interactive version of the GoaL-E algorithm that enables end-to-end LfD for non-expert teachers. With this final algorithm, the robot is interactive by communicating the learned skill through execution and by verbalizing the monitoring result. In addition, the teacher is able to provide feedback in robot's skill executions, including the sampled ones. Interactive GoaL-E has been evaluated with 12 participants in *Experiment IV*. The results show that teacher feedback can increase action and goal model performance and non-expert teacher data can be used in self-improvement.

1.3 Outline

The remainder of this document is organized as follows

- **Chapter 2 – Related work** provides a survey of prior research and positions this thesis in relation to the existing literature.
- **Chapter 3 – Preliminaries** presents the robot platforms and the environmental setup shared by the experiments. In addition, it describes a generic protocol for conducting human-robot interaction studies for skill learning in the context of learning from demonstration. The experiments performed for this thesis follow this protocol.
- **Chapter 4 – Learning from Everyday People** presents findings and observations about non-expert people teaching robots from multiple user studies. This chapter also introduces keyframes, hybrid demonstrations and describes the observations that motivate goal learning.

- **Chapter 5 – Keyframe Based Learning from Demonstration** develops a method to learn from trajectory, keyframe and hybrid demonstrations in a unified way by converting all the input to keyframes while retaining velocity and acceleration information.

- **Chapter 6 – Action and Goal Models** introduces action and goal models, presents the learning approach and describes how these models are used for execution and monitoring.

- **Chapter 7 – Action and Goal Learning with People** presents the findings of a user study that looks at the execution and monitoring success of the goal models when trained from batch demonstrations of naïve teachers.

- **Chapter 8 – Self-Improvement** describes the self-improvement procedure, introduces adaptive sampling and presents the evaluation of this method with expert user demonstrations.

- **Chapter 9 – Interactive Learning** closes the interactive loop of the developed approach and presents its evaluation with naïve users.

- **Chapter 10 – Conclusion and Future Work** discusses the contributions of this thesis, directions for future research, and concluding remarks.

CHAPTER II

RELATED WORK

General surveys of the field of LfD can be found in [14, 11, 24]. In LfD, existing works tend to fall into two categories, what we call skill learning and task learning. Skills are defined as low-level short duration actions (*e.g.* pouring) whereas tasks are higher level partially ordered combination of such skills (*e.g.* making breakfast). This thesis mainly concerns learning skills, however, it also aims to bridge this dichotomy, by learning task goals for a skill learning method. In addition to skill and task learning, this thesis involves topics not typically addressed in LfD work: skill monitoring, multiple demonstration modalities and human-robot interaction. The following sections present existing literature in relation to the work in this thesis.

2.1 Keyframes and Trajectory Segmentation

Traditional LfD techniques work with *trajectory demonstrations*; demonstrations that are continuous sequences of points in the state space. The start and the end of a trajectory are often indicated by the teacher, and the robot records (with a sufficiently high frequency) the change of the state between these two events. In contrast, *keyframes* are sparse (in time) set of sequential points that the teacher demonstrates to the robot. In this thesis, keyframes are used extensively.

Keyframe related ideas exist in other fields as well. In industrial robot programming, these are referred to as *via-points*. A robot programmer records important points by positioning the robot and specifies how to move between them. Keyframes are used extensively in computer animation [57]. The animator creates frames and specifies the animation in-between. However, there is no learning component in these approaches. In [51], keyframes are extracted automatically from continuous demonstrations and updated to achieve the

demonstrated skill. Another approach is to only record keyframes and use them to learn a constraint manifold for the state space in a reinforcement learning setting [16]. Whole body grasps for a simulated humanoid are learned in [37] by forming template grasp demonstrations via keyframes, which are the start/end points of a demonstration, the points of contact and points of lost contact with the objects.

A related topic to keyframes in LfD is trajectory segmentation. Keyframes are similar to the segmentation points. In addition, this thesis introduces *hybrid demonstrations*, in which teachers can demonstrate both keyframes and trajectories in any sequence and number in the context of a single demonstration. This gives the teacher the ability to segment trajectories as well as to provide keyframes where the trajectory information does not matter.

Some LfD methods employ automatic trajectory segmentation to break down demonstrations in order to facilitate learning. These include [45, 54] that are mentioned in the Sec. 2.3. Although similar, trajectory segmentation and keyframes are not directly comparable. Some of the trajectory segmentation methods aim to break down trajectory demonstrations to learnable chunks to alleviate model selection, and as such the segmentation points may not be of specific importance. Other trajectory segmentation methods try to find important points (*e.g.* when the relevant reference frame changes during demonstration) to gain higher level information but this is a hard task and it depends on a predetermined definition of this importance. In contrast, the aim of keyframes and hybrid demonstrations is to let the human teacher highlight the important parts of the skill from his/her perspective.

2.2 Skill Learning

2.2.1 Supervised Skill Learning

There is a large body of literature on learning motor control models, or *skill learning*. Dynamical system approaches such as Stable Estimator of Dynamical Systems (SEDS) [40] and Dynamic Movement Primitives (DMP) [58] as well as mixture models, such as

Gaussian Mixture Models [21], are skill learning methods. These methods all involve estimating parameters of a dynamical system, a distribution and/or a function approximator. In contrast, the method described in [64] is a non-parametric approach to learning from demonstration. The current point cloud is mapped to previously seen point clouds via non-rigid registration and the best mapping is applied on the corresponding trajectory which is then executed on the robot. These methods work with trajectory demonstrations. This thesis makes extensive use of keyframes and as a result, these methods are not directly applicable.

Classification based methods, such as [35, 39], can be seen as supervised policy learning methods in which input demonstrations are mapped to action primitives or robot states. Then the transition structure (*e.g.* order of actions or transition probabilities within states) represent the skill. The same idea can be used to learn tasks as well. The work presented in this thesis does not have any pre-determined symbols to be mapped to actions or sensory states.

2.2.2 Reinforcement Learning

Some difficult to program skills have easy to represent goals or cost. In such cases, a reinforcement learning (RL) approach is viable. However, traditional RL methods do not scale well with high number of dimensions and continuous spaces such as those encountered in robotics. Within reinforcement learning, policy search methods have been shown to be suitable for skill learning for robots with high number of degrees-of-freedom (dof). In most of these methods a potentially unsuccessful initial policy is learned from demonstrations, which is then input to the policy search method along with the reward function. Surveys for RL in robotics can be found in [41, 27]. These methods require a pre-determined reward/cost function that describes the goal of the skill. Everyday people are very unlikely to come up with such functions to make these methods work. The self-improvement part of this thesis does not need a reward function to be programmed by the teacher and instead

learns the goal from demonstrations.

2.2.3 Inverse Reinforcement Learning

Another approach to skill learning is Inverse reinforcement learning (IRL) [1] or similarly inverse optimal control (IOC) [61]. In IRL, a reward or cost function is estimated from demonstrations and then used to extract a policy. The main idea behind the IRL approaches is that the reward function is a better representation of the demonstrated skill than the policy.

The goal learning idea is similar to the main idea of inverse reinforcement learning (IRL); extracting a reward function from demonstrations. However, the IRL methods cannot be directly applied and there are some problems to overcome. Keyframes provide sparse rewards which is not typical for IRL. The dimensionality of our goal space is prohibitive for some of the existing IRL methods to be used in interaction time [1]. The IRL idea was developed with expert[2] demonstrators in mind. Naive teacher input to IRL methods has been considered as important but has not been tackled. The work in [65] attempts at minimizing inconsistent and noisy demonstrations by weighted regression (based on robot's ability to achieve them), averaging and removing demonstrations. However, this approach is still geared towards teachers that are expert at using the robot. While these issues are not insurmountable for IRL, they would be additional research that is not the focus of this thesis.

2.3 Task Learning

The aforementioned sections dealt with learning low level skills. This section will highlight a few task learning methods from a vast literature that is relevant.

The main idea of *task learning* methods is to map motor and sensor level information to pre-defined symbols and learn the relationship between them [47, 53, 29, 26]. Typically

[1] Fast enough to have a fluid interaction with the teacher

[2] Expert in the sense of demonstrations, not necessarily the underlying algorithms.

these approaches assume a pre-defined mapping of sensor data to objects/symbols, and assume the task uses a given set of primitive actions. In contrast, the work in this thesis aims to learn the perceptual goals of a skill without assuming predefined symbols for actions or objects. Only the features extracted from sensors and object segmentation method is pre-programmed.

In [56], an incremental approach is taken to learn a knowledge base for household tasks from human demonstrations. In [53], the pre- and post- conditions of behaviors are learned and encoded within a behavior network. In [29], ordering constraints between actions are learned through multiple demonstration in addition to modeling placement locations (*i.e.* goals). In [26], a discrete finite automaton is learned from human demonstrations for assembly tasks (described by a formal language), and it is coupled with pre-defined robot motions to constitute a *motion grammar*.

Kulic *et al.* describes an incremental LfD system in [45]. The demonstrations are first segmented and the resulting trajectory segments are treated as small actions which are then learned using Hidden Markov Models (HMM). Then these models are grouped together with the similar models to form a tree structure. In addition, a graph is built between action groups to form a higher level representation. A similar approach is presented in [54], which uses DMPs to learn small actions. These methods also aim to bridge the gap between *skill learning* of particular actions and *task learning* of higher level plans or graph structure. An example LfD system towards this direction is presented in [55], which builds a finite state representation of the task through demonstrations and leverages this for adaptive skill sequencing and error-recovery. Their work differs from this thesis work since they are not building perceptual models for monitoring the execution.

2.4 Goal Learning and Skill Monitoring

In most of the existing work, goals or the structure is learned from pre-defined symbols (in case of tasks) or encoded in a cost function (in case of reinforcement learning). In contrast,

the work in this thesis aims to capture this by learning goal models through demonstrations. A similar approach is described in [38], which presents a method that learns kinematic task constraints through low-level demonstrations and generalizes them with semi-supervision. The aim is to learn these constraints for planning. The demonstrator, environment and objects are fully modeled. The thesis work focuses on simple interactions with naïve teachers rather than heavily instrumented expert demonstrators. Another similar work is presented in [22]. Pre- and post-conditions for pick and place skills are learned from continuous perceptual features from human demonstrations. The aim is to use the learned models to bootstrap learning new skills (in the form of pre- and post-conditions). The insight of this thesis is that the salient keyframes provided by the teacher are highly suitable for learning perceptual representations of goals, similar to [22], but taking intermediate steps into account. In addition, this thesis work learns the accompanying action as well as the goal and uses this to monitor the executed action, in contrast to using the learned goals to bootstrap learning of new skills.

Another similar idea related to monitoring and presented along skill learning is [59]. The robot executes its learned skill, collects sensory data and the successful executions are labelled by hand. The robot then builds a Gaussian model for the trajectories for each sensory state dimension, which requires a high number of skill executions. These models are used in hypothesis testing during future executions to monitor the skill. Further related ideas are presented in [60]. In contrast, the work in this thesis uses the sensory data obtained during the skill demonstrations to learn a goal model without further manual labelling and skill repetition.

The IRL/IOC models (see Sec. 2.2.3) can also be considered as a form of goal learning, since they are representing the goals implicitly within the learned cost function. For example, the low cost-to-go (or high value) regions can be considered as desirable (*e.g.* as sub-goals or goals).

There is resemblance between gesture recognition (*e.g.* [66]) and skill monitoring with

goal models. In both, a model that is trained on observations is used to recognize future observations. In gesture recognition, the aim is to find the right gesture, or analogously, find which skill is being executed from observation. In skill monitoring, the robot knows what it is executing and monitors whether it is achieving that or not. The monitoring problem is easier than gesture recognition. However, the state spaces in gesture recognition are mostly dependent on the type of gesture that is being recognized (*e.g.* for body gestures the estimated pose of the human body). In skill monitoring, our aim is to use a generic state space to encode a variety of object related task goals.

2.5 Human Robot Interaction and LfD

One of the primary distinctions of the work in this thesis is the focus on Human-Robot Interaction in the context of LfD. For example, most methods implicitly assume the demonstrators are good at using the robot. In a large number of prior work, the demonstrators used in evaluations are the developers of the methods themselves.

The related work on LfD evaluations with non-experts is sparse. Calinon *et al.* highlight the importance of teachers in LfD in [19] but do not perform a user study. Suay *et al.* , in [67], tested three LfD methods with naïve users. One of their findings is that these users were not able to teach successful policies whereas experts (the authors themselves) were able to do so within minutes. Thomaz *et al.* have investigated how humans teach software agents and robots various tasks, and developed a reinforcement learning algorithm to leverage natural human behavior [68, 69]. In the context of skill learning, Cakmak and Thomaz investigated how naïve users can be guided to provide better demonstrations through teaching heuristics and active embodied queries by the robot in LfD interactions [17].

In [73], kinesthetic teaching is embedded within a dialog system that lets the user start/end demonstrations and trigger reproductions of the learned skill with verbal commands. In another study [48], four types of force controllers that effect the response to

users are evaluated for kinesthetic teaching. The study addressed human preferences on which controller was the most natural. In [44], a method of teaching stiffness during a skill execution is presented and tested with non-expert people.

2.6 Incremental and Interactive Learning

An interactive approach to LfD in which the robot communicates its learning state somehow (*e.g.* by executing the learned skill) during the interaction requires online learning of the skill. In some of the aforementioned methods, the data is first collected and learning is done afterwards which is also called batch learning. An alternative approach to batch learning is called incremental learning. In incremental learning, the models are updated as new data comes in. Some example incremental learning approaches are presented in [45, 54, 18].

In [23], a more interactive approach is taken. The robot asks for demonstrations at states that it is not confident on. In addition, the human has the ability correct the robot's mistake with additional demonstrations. Cakmak *et al.* , in a body of work presented in [17], lets the robot ask questions to the teacher beyond the demonstrations (*e.g.* about the features) during the teaching interaction.

Although researchers have acknowledged the advantage of interactive learning and the potential benefit of taking the teacher into account, there are not many studies that actually test these on real users. To the best of my knowledge, there are no experiments that compare batch vs incremental learning in the context of robotic LfD. The closest study is done by Zang *et al.* with a software agent in [74]. They found that simply by watching the learner execute the learned action, naïve teachers provide better demonstrations (as compared to the batch learning case) as the LfD interaction progresses and the resulting agent performance increases as a result.

2.7 Data Representation and Demonstration Modalities

In LfD, demonstrations are often represented as arm joint trajectories and/or end-effector path [20, 33]. Some also consider the position of the end-effector with respect to the target object of the skill [15, 32]. Typically start and end points of a demonstration are explicitly demarcated by the teacher. Most studies subsample the recorded data with a fixed rate [10, 15]. Demonstrations are often time warped such that a frame-by-frame correspondence can be established between multiple demonstrations [33]. Some methods also use forces/torques as their input [43, 62].

Most current work does not incorporate the perceptual state into the learning problem other than object pose estimation. In this thesis, features extracted from a sensor are used during learning.

Typically in LfD, the problem is defined as learning in the motor space or in a state space that is known in advance to be good for representing a particular skill. For example in [59], the state space to learn a pool stroke were highly specialized to that skill alone (*e.g.* cue rotations, tip offset, elbow posture). Other seminal examples with a skill specific feature space include the pendulum swing up and balance task [12], playing table tennis [52], flipping pancakes [42] and flying a model helicopter [2]. In all of these prior works automatic feature selection problem is acknowledged as an important problem for future work. However, it is neither feasible to assume a specific state space (perhaps other than the end-effector or joint space) for skills that people want to teach their robots nor to assume that people will be able to identify appropriate state spaces and transfer these to the LfD algorithm. In contrast to the existing work, this thesis uses a generic and high-dimensional feature space for the goal models and the space of end-effector poses with respect to the target object for the action models. An interesting work, which is not the focus of this thesis, is to learn this feature space as well, *e.g.* as in [30].

There is a vast range of different input schemes that lead to very different interactions

for the teacher: teleoperating a robot, performing a task in a motion capture setting, performing the task uninstrumented, physically guiding to robot *etc.* The last one is called *kinesthetic* teaching.

In this thesis, and many of the aforementioned skill learning works such as [10, 33], kinesthetic teaching is the primary mode of demonstration. In kinesthetic teaching, there is not a correspondence problem and demonstrations are restricted to the kinematic limits of the robot (*e.g.* workspace, joint limits). Moreover, extra hardware/instrumentation, such as motion capture or teleoperation devices, is not necessary.

Teleoperation is also used at some points in this thesis. Teleoperation has been used in robotics for more than 50 years, with the focus on dealing with delays, information loss, instabilities, operator noise, telepresence etc. [34]. It is also used in the context of LfD, *e.g.* a method which injects haptic information to guide the user for better demonstrations is presented in [36].

CHAPTER III

PRELIMINARIES

The experiments performed in this thesis follow similar protocols and experimental setups. In this chapter, I will describe a typical experimental protocol and some elements of the experimental setups. Any experiment specific parts will be presented in their respective chapters.

3.1 Setup

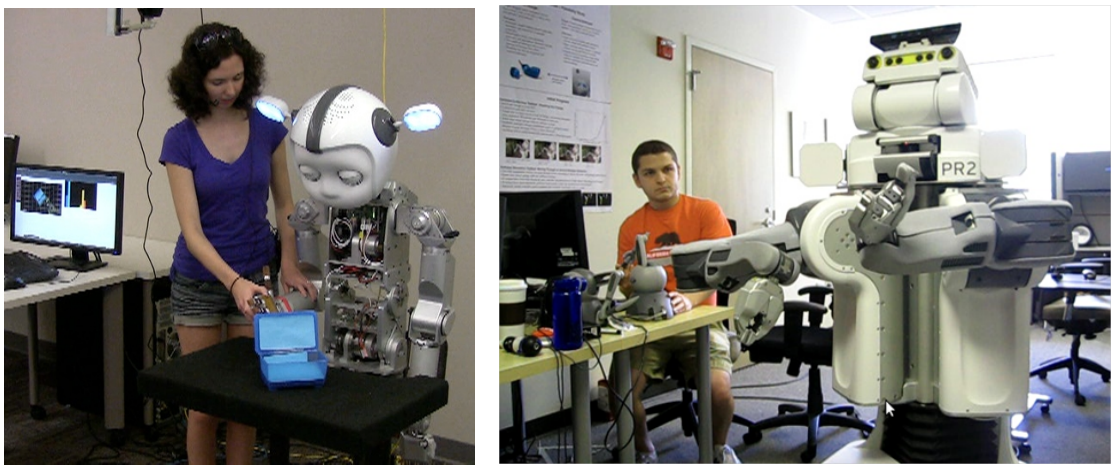
3.1.1 Demonstration Modalities

In this thesis, the majority of the demonstrations are from kinesthetic teaching. In kinesthetic teaching, the teacher guides the robot's arm physically to provide demonstrations, as shown in Fig. 3(a). In one experiment, teleoperation was used to provide demonstrations, as depicted in Fig. 3(b). In teleoperation, a 6-dof Phantom Omni device was used to directly control the end-effector of the robot.

3.1.2 Robots

In this thesis, three robots were used for experiments:

- *Simon*: An upper-torso humanoid robot with two series-elastic actuated 7-DoF arms, two 4-DoF hands, and a socially expressive head and neck, including two 2-DoF ears with full RGB spectrum LEDs [28]. The arm kinematic configuration is $3-1-3$; a spherical shoulder followed by an elbow which in turn is followed by a spherical wrist. This kinematic configuration is very similar to the human arm configuration. Simon can be seen in Fig. 4(a).



(a) Kinesthetic teaching (b) Teleoperation

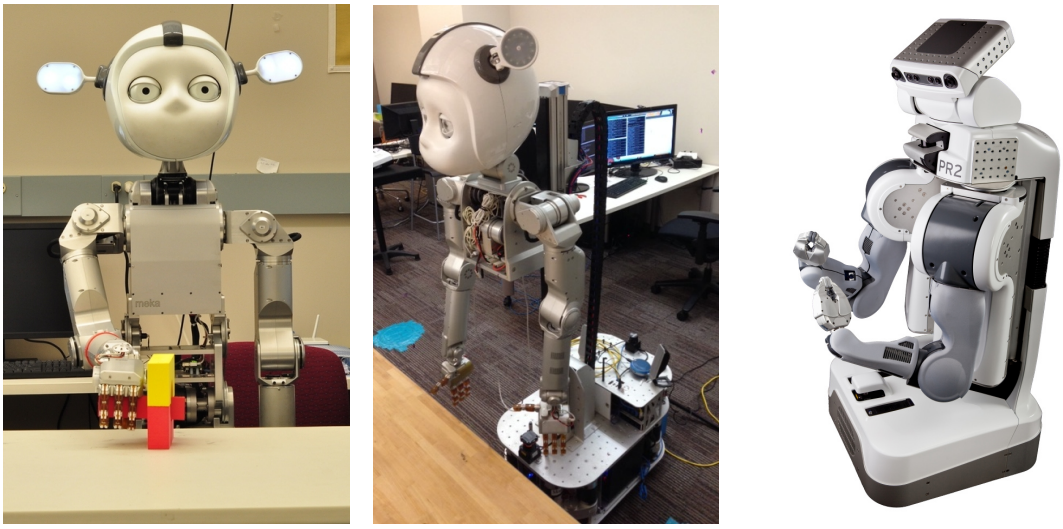
Figure 3: Two input modalities for LfD.

- *Curi*: A mobile manipulator with two series-elastic actuated 7-DoF arms, two 5-DoF hands, and a socially expressive head and neck. In addition, the torso is on a vertical linear actuator to afford greater manipulation workspace. Curi is very similar to Simon and can be seen in Fig. 4(b).

- *PR2*: A mobile manipulator with two 7-DoF arms, gripper hands and an articulated neck. The arm kinematic configuration is $3-1-2-1$; a spherical shoulder followed by an elbow which in turn followed by a roll-pitch-roll wrist. PR2 can be seen in Fig. 4(c).

These robots have several common functionalities that are utilized in the experiments. Kinesthetic teaching requires the robot to counteract the gravity so that the teacher can easily manipulate the robot's arm. This is called *gravity compensation*. Simon and Curi have active gravity compensation. The torques necessary to counteract the gravitational forces at each joint are calculated by using a mass model of the arm. PR2 has passive gravity compensation, that counteracts gravity using springs.

The robots' hands are used for *power grasps* that aim to encapsulate the object but not for precise manipulation. Power grasping is done by closing the fingers until an object is grasped. This allows the robots to grasp objects of varying sizes. PR2 uses touch sensors on



(a) Simon (b) Curi (c) PR2

Figure 4: The robots used in this thesis.

its gripper to understand whether it grasped an object or not. Simon and Curi does not have touch sensors but they can measure the approximate torque in the fingers. For grasping, the fingers are closed until the measured torques reach a certain value.

3.1.3 Speech Commands

Speech commands are used to move the interaction forward and provide demonstrations (*e.g.* to mark keyframes). The list of common speech commands are provided in Table 1. Any additional commands will be provided in their respective chapters. We use *Microsoft Windows 7 Speech API* for speech recognition.

3.1.4 Environment

All of the experiments and evaluations done in this thesis include teaching skills on a table top. The robot is positioned in front of a table. If the teacher is kinesthetically teaching the robot, he/she stands next to the robot arm used in teaching. This side is kept clear for participant comfort (*e.g.* Fig. 3(a)). For some of the experiments, an overhead ASUS Xtion Pro LIVE (RGBD camera) is employed to get object information with an overhead view

Table 1: Speech commands for controlling the interaction and their function

Command	Function
Let's begin the experiment	Start the interaction
Release your arm	Activate gravity compensation mode
Hold your arm	Activate joint position mode
Let's learn a new skill	Switch to next skill
Like this	Start trajectory
That's it	End trajectory
Start Here	Mark the first keyframe
Go Here	Mark an in-between keyframe
End Here	Mark the last keyframe
Close your hand	Close the robot's fingers
Open your hand	Open the robot's fingers

of the tabletop. The Point Cloud Library (PCL)[1] is used to process point cloud data. The details of how to process this data is explained in Sec. 6.2.1.

3.2 Experimental Protocol

The experiments performed in this thesis compare two or more teaching conditions or show that the developed algorithms work with naïve teacher data. All the experiments include teaching multiple skills. A systematic way to handle these experiments is needed to ensure that the data is as unbiased as possible. This section introduces the experimental protocol developed to perform skill learning from demonstration experiments with non-expert teachers. The user studies presented in this thesis utilize the following protocol.

1. Greet and thank the participants. Tell them that they'll be teaching skills to a robot.
2. Ask them to sign the consent form
3. Introduce the robot, modality(s) of teaching and the sensors if applicable
4. Introduce the generic speech commands
5. Introduce the next condition based on counterbalancing

[1] `http://pointclouds.org/`

6. Optional: Free-style practise with the condition
7. Introduce the specific speech commands
8. Optional: Demonstrations of a practise skill in the current condition
9. Next skill based on counterbalancing
    (a) Describe the goal of the skill verbally
    (b) Let the participant demonstrate the skill
    (c) Optional: Let the participant watch the robot execute
    (d) Repeat b until done
10. Repeat 9 until all skills are done
11. Condition Survey (if applicable)
12. Next condition if applicable
13. Repeat 5 until all conditions are done
14. Exit survey (if applicable)
15. Thank the participant for his/her time.

There are a few important points in evaluating skill learning with everyday people. They will not know anything about the robot and the experiment. They will not have any experience in providing demonstrations to the robot. In addition, the teachers will need to use speech commands. It is important to get them familiarized with the interaction, the robot, the commands and the experimental setup so that the *learning effects* are minimized. These effects arise from the fact that the teachers will get more experience as they interact with the robot. The initial experience changes the way participants teach by a significant amount. A practice session alleviates this such that the participant starts the experiment

with a certain level of experience. Therefore, it is important for the participants to practice the way they provide demonstrations and the way interaction works before the experiment starts collecting data. Since a practice session cannot entirely eliminate the learning effects, it is important to *counter-balance* the order of the conditions and the skills. Another effect is *participant fatigue*. The participants get tired or lose interest after interacting with the robot after a while, leading to bad data. The experiments in this thesis keep the interaction under 30 minutes if the aim of the thesis is bulk data collection as this will lead participants to lose interest. If the experiment is interactive (*e.g.* robot executing skills), it is allowed to be 60 minutes since an interactive robot is more engaging. The 60 minute upper limit is selected to prevent the participant getting tired during the interaction.

The experiment starts with introduction to the robot and the experiment. The participants are told about the way that they will teach the robots. The conditions are experiment specific. These mainly include the type of demonstrations participants provide. For example in Experiment I, described in Sec. 4.1, these conditions are keyframe and trajectory demonstrations. Not all the experiments have different teaching conditions such as the one described in Chapter 7. Practice usually involves two parts. The first part includes moving the robots arm to pose it in canonical configurations. These configurations are chosen such that the workspace and joint limits of the arm are highlighted. The second part includes a practice skill to get the participant acquainted with the specific condition. For example, this includes demonstrating how to place an object in a certain location using keyframes.

The skills are explained verbally to the participant. The reason not to physically show the skill is to prevent biasing the participant on how to demonstrate it. If the experiment is interactive, the participant is allowed to see what the robot has learned so far. In some experiments, the object locations are varied systematically so that a wide-range of demonstrations are collected. If the participant decides to stop teaching or that another stopping condition (*e.g.* number of demonstrations) is met, the participant moves on to the next skill. If all the skills are done for the current condition the participant can be asked to do a survey

if applicable. The reason for having a survey after each condition is to prevent biasing of the results with other conditions. After this, the participant moves on to the next condition. When all the conditions are done, the experiment finishes with an optional exit survey.

There are parts of the protocol that are experiment specific. These are:

- Hypotheses and Research Questions
- Metrics for evaluation: Skill execution success and Likert-scale survey questions are examples for used metrics
- Number of participants
- Teaching conditions
- Skills that are taught to the robot

CHAPTER IV

LEARNING FROM EVERYDAY PEOPLE

The main goal of this thesis is to enable non-expert people to teach skills to robots. Towards this end, the research takes a Human-Robot Interaction perspective on Learning from Demonstration. The work in this chapter concentrates on learning actions interactively from naïve teachers. The LfD system that underlies the experiments presents in this chapter can be seen in Fig. 5.

The main questions arise from taking an HRI approach are the following:

- How should a non-expert provide demonstrations?
- What does non-expert data look like?
- How can we learn from non-expert teacher data?

In the field of LfD, *trajectory* demonstrations are the standard way of getting demonstrations from teachers. In contrast, factory robots are generally programmed with *keyframes*[1]. There is no study that explores which one would be best suited for non-expert teachers in the context of LfD. This chapter starts by describing a user study on comparing *keyframe*

[1] Mainly referred to as *via-points*

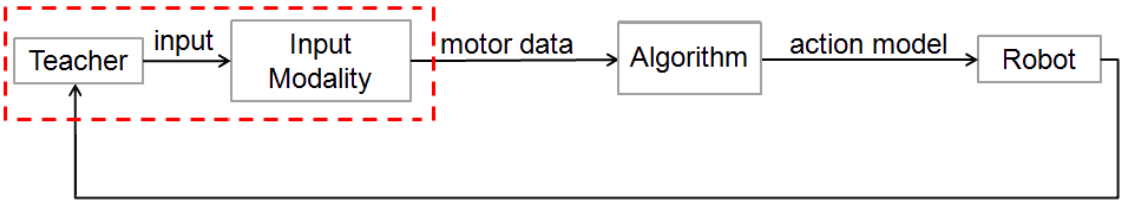


Figure 5: The version of the developed system used for HRI studies on LfD for action learning. The red box with dashed lines highlights the main focus of this chapter.

and *trajectory* demonstrations with kinesthetic teaching and analyze the results in detail. Based on the outcomes of this experiment, this chapter introduces the *hybrid demonstrations*; demonstrations which can have arbitrary keyframe and trajectory demonstration segments. Then, it presents a pilot study on comparing *kinesthetic teaching* and *teleoperation*. Finally, it presents an experiment on keyframe, trajectory and hybrid demonstrations on teleoperation. Based on the results of these experiments, the necessity of developing a LfD method to learn from hybrid demonstrations and goal oriented nature of naïve users will be discussed. The work presented in this chapter is published in [4, 6].

4.1 Experiment I: Trajectories versus Keyframes

In a typical LfD interaction, each demonstration is an entire state *trajectory*, which involves providing a continuous uninterrupted demonstration of the skill. This thesis explores the alternative of providing a sparse set of consecutive *keyframes* that achieve the skill when connected together. This section presents an experiment with kinesthetic teaching that compares these through quantitative measures, survey results and expert evaluations. The results suggest that both types of demonstrations are suitable for kinesthetic teaching from the user's perspective and both communicate different information. Two modified keyframe interactions are also introduced and their utility evaluated.

There is a lack of user studies in the field of LfD. It is largely unknown whether non-expert teachers would be able to provide demonstrations for existing LfD methods to work. The experiment presented in this section, Experiment I, is among the first user studies that explicitly evaluate LfD with non-expert users. As stated previously, the way that non-expert teachers provide demonstrations and how the resulting data looks like are open questions. This experiment aims to compare various ways of providing demonstrations to the robot and their effect on certain skill types with non-expert teachers. The specific research questions that this experiment answers are presented in Sec. 4.1.2.3.

4.1.1 Demonstration Methods

The experiment explores three different ways for teachers to demonstrate skills with kinesthetic teaching: *trajectory demonstrations*, *keyframe demonstrations*, and *keyframe iterations*. The number of demonstrations are left to the teacher. The motor data used for this experiment is the seven joint angles of the robot's right arm. The experiment is interactive; After each demonstration, the teacher can have the robot perform the current state of the learned skill and adjust his/her demonstrations accordingly. The teacher uses speech commands to facilitate the interaction. The robot Simon is used in this experiment.

4.1.1.1 Trajectory Demonstrations

The teacher is informed that the robot will record all the movement they make with its right arm. The teacher initiates the demonstration, moves the arm to demonstrate to the robot how to perform the skill and finishes.

The off-the-shelf Gaussian Mixture Model (GMM) based LfD method described in [21] is used to learn the action model. Similarly, Gaussian Mixture Regression (GMR) is used to execute the skill.

4.1.1.2 Keyframe Demonstrations

The teacher is informed that the robot will only record the arm configuration when they mark a keyframe, and it will not record any movements between these keyframes. The resulting data from this interaction is a sparse trajectory of joint angles.

The learning is the same as the previous approach but the execution is different since the clusters obtained from GMMs are of different nature. With keyframes, the clusters correspond to start and end of linear segments of the skill, whereas with trajectories, they are more likely to be mid-points of these segments. To execute the action, splines are fit between the cluster means by assuming zero velocity and acceleration at each cluster. Timing information for each keyframe is generated by using a constant average velocity.

Table 2: Additional Speech commands for Keyframe Iterations

Command	Function
Next frame, Previous frame	Navigate through current demonstration
Modify this frame	Change the arm configuration of the current keyframe
Delete this frame	Delete the current frame
Play current demonstration	Play the current demonstration being iterated
Record this demonstration	Submit the current demonstration to the learning set

4.1.1.3 Keyframe Iterations

The experiment also explores an augmented version of keyframe demonstrations, in which a new demonstration is an iteration of the current learned skill.

In this mode, an initial demonstration is provided using keyframe demonstrations. Then the teacher can navigate through and edit the frames of this demonstration to create additional demonstrations. Learning is the same as in keyframe demonstrations. The teacher uses the additional speech commands described in Table 2 during this interaction. The initial keyframes for the following iterations come from the means of the learned clusters.

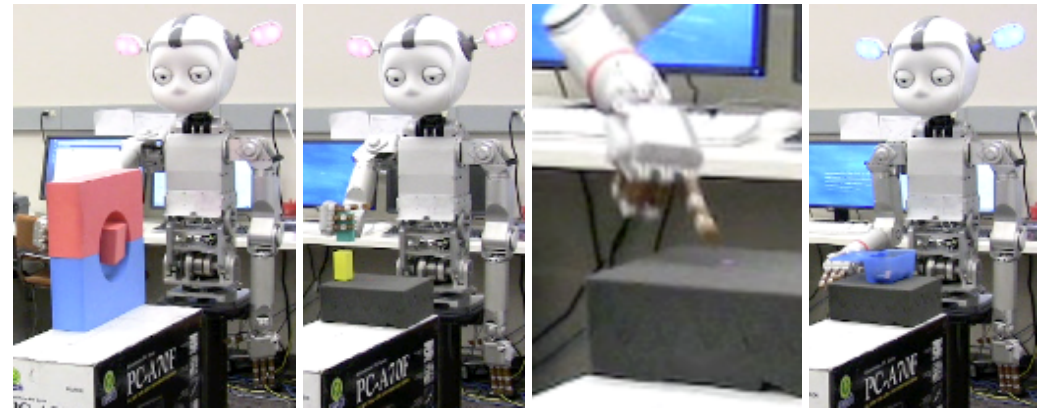
4.1.2 Experiment Details

The experiment follows the protocol described in Sec. 3.2. The experiment specific details are provided in this section.

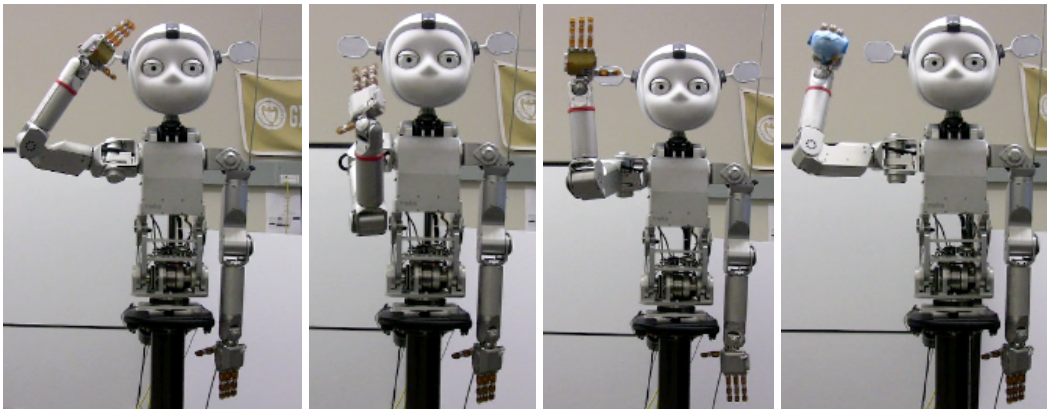
4.1.2.1 Skills

This experiment differentiates between two types of skills. *Goal-oriented* skills are related with achieving a particular world state (e.g., finger tip on a point while avoiding obstacles.) *Means-oriented* skills, on the other hand, include a gesture or communicative intent. Four skills are taught for each type as depicted in Fig. 6.

The goal-oriented skills are as follows. (Fig. 6(a-d)). **Insert**: insert the block in hand through the hole without touching other blocks. **Stack**: stack the block in hand on top of another block on the table. **Touch**: touch a certain point with the finger tip. **Close**: close the lid of a box without moving it.



(a) Insert (b) Stack (c) Touch (d) Close



(e) Salute (f) Beckon (g) Raise (h) Throw

Figure 6: Goal-oriented (a-d) and means-oriented (e-h) skills.

There is a single goal position per skill for the ease of the experiment. Multiple goal positions would prolong the experiment and the aim is not to analyze the generalization properties of the methods but rather to analyze the utility of the keyframe demonstrations from a user's perspective.

The means-oriented skills are as follows. (Fig. 6(e-h)). **Salute**: perform a soldier's salute. **Beckon**: perform a gesture asking someone to come closer. **Raise-hand**: raise the robot's hand as if it is asking for permission. **Throw**: perform a throwing gesture with a ball (without actually releasing the ball).

4.1.2.2 Conditions

The experiment has four conditions, and uses a within-subject design, *i.e.* all the users participated in all the conditions. Three conditions correspond to the teaching methods in Sec. 4.1.1. In addition, a fourth condition tests the effect of the initial demonstration in keyframe iterations.

- *Trajectory Demonstrations (TD):* Participants give one or more trajectory demonstrations for each skill.
- *Keyframe Demonstrations (KD):* Participants give one or more keyframe demonstrations for each skill.
- *Keyframe Iterations (KI):* Participants use keyframe iterations to teach the skills.
- *Keyframe Adaptation (KA):* Participants start with a predefined, slightly failing skill (e.g. touch is off by a few centimeters), instead of giving her/his own initial demonstration. They use the KI interaction to improve this skill.

The participants first teach the robot in the TD and KD conditions. The order of these two is counterbalanced. After these two conditions, KI and KA conditions follow. Thus the order was $(TD \mid KD) \rightarrow KI \rightarrow KA$. This ordering is to prevent biasing a participant in the

KI condition with the predefined skill used in the KA condition. The type of skill taught to the robot across the TD and KD conditions are varied, each participant taught one *means-oriented* skill, and one *goal-oriented* skill in these modes. Only *goal-oriented* skills were used in KI and KA conditions, to reduce experiment duration.

At the beginning of each condition, participants taught a *pointing* skill to the robot for familiarization and practise with the condition. Participants were also allowed to move the robot's arm to practice before recording demonstrations.

4.1.2.3 Research Questions

This experiment aims to address three research questions:

Q1 When everyday people teach the robot, what are the effects of each demonstration type?

Q2 Does the teaching method have any effect on learning different types of skills?

Q3 Can simple extensions to keyframe demonstrations (iteration and adaptation) increase performance/preference?

For Q1, effect of demonstration type, TD and KD conditions are compared. For Q2, learning of different skill types, *goal-oriented* and *means-oriented* skills are compared in TD and KD conditions. For Q3, effect of extensions to keyframes, first KD and KI are compared, then KI and KA are compared.

4.1.2.4 Metrics

Two different methods were used to measure the quality of the different types of learned skills. The *goal-oriented* skills are evaluated with three levels of success criteria, and the *means-oriented* skills with expert ratings.

The performance of goal-oriented skills were scored separately by the two of the experimenters, using three levels of success criteria: Success-PartialSuccess-Fail. The scoring

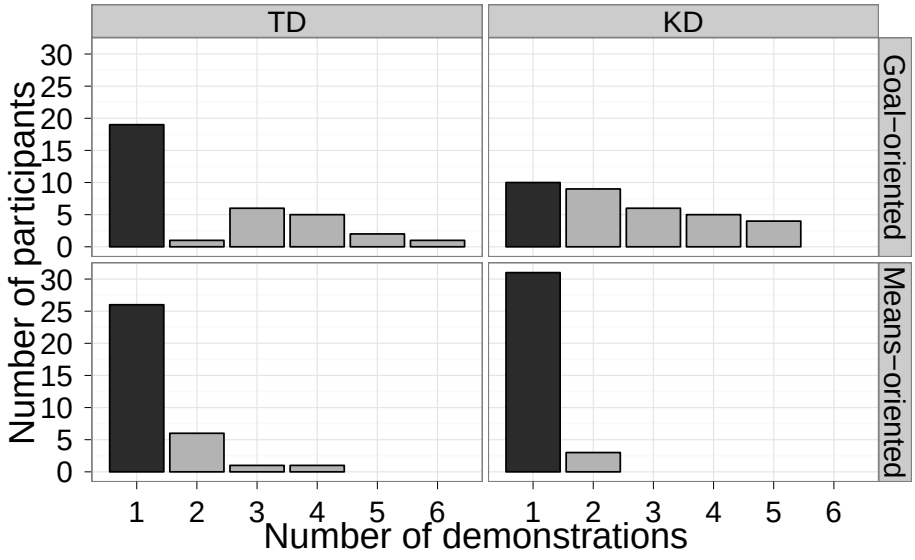


Figure 7: Histogram of number of demonstrations provided by participants in KD and TD conditions.

was based both on the recorded videos of the experiment and on the skill performances recreated on the robot. In the few cases where there was disagreement, the two coders revisited the example and reached a consensus on the scoring.

Unlike the goal-oriented skills, success for *means-oriented* skills is subjective. Therefore, expert ratings of the recreated movements were used to evaluate the performance. The experts, whose specialities are in computer animation, were asked to answer three 7-point Likert-scale questions (see Fig. 9) for all means-oriented skills taught by all participants [2]. The questions were about *appropriate emphasis*, *communicating intent*, and *closeness to perfection*.

To evaluate the user's preferences, 7-point Likert-scale questions were used. These questions were administered after each condition, about *Feel*, *Naturalness*, *Ease*, and *Enjoyability*. Open-ended questions were also asked after the first two conditions and at the end of the experiment.

The number of demonstrations, the number of keyframes, the time stamps for every

[2]The experts were compensated with $25 for their time

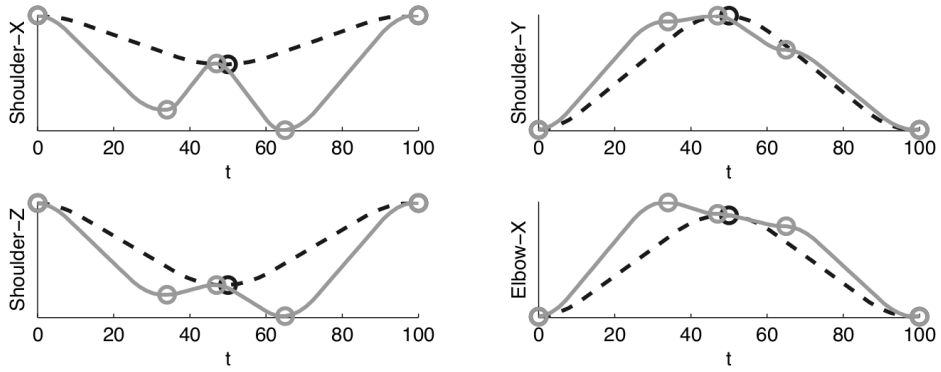


Figure 8: An example, in the KD condition, of forgetting obstacle avoidance keyframes in a first demonstrations (dashed line), and providing them in a second (solid line) while teaching the *Touch* skill.

event, and all trajectories of the joint movement during demonstrations were also measured as metrics to give information about the overall kinesthetic interaction.

4.1.3 Results of Experiment I: Trajectories versus Keyframes

The study had 34 participants (6 females, 28 males between the ages of 19-47), who were undergraduate and graduate Georgia Institute of Technology students with no previous machine learning and robotics experience. 22 of the participants taught Simon in all four conditions, while 12 only performed the first two conditions [3]. A single experiment took on the order of one hour.

4.1.3.1 Trajectory vs Keyframe Demonstrations

First the TD and KD experimental conditions are compared and five observations are made. The observations reported in this section did not vary across particular skills.

Single demonstrations are common: Users were able to see what the robot has learned after each demonstration and either decide to move on or give another demonstration. Fig. 7 shows the number of demonstrations provided by participants in TD and KD. It can be seen that teaching with a single demonstration was common in both modes. For goal-oriented skills, a larger portion of the participants provided a single demonstration in the

[3]The participants were compensated with $10 for their time

TD condition than in the KD condition (19 versus 10). It was common in the KD condition to forget to provide keyframes that allow the robot to avoid obstacles while trying to achieve the goal. These frames were provided by participants in subsequent demonstrations after observing the performed skill colliding with obstacles (*e.g.* see Fig. 8). For means-oriented skills, teaching with a single demonstration was more common in the KD condition than in TD (31 versus 26).

Trajectory demonstrations may be better for teaching goal skills in a single demonstration: Table 3 provides the distribution of participants according to the success of the goal-oriented skills they taught [4]. More participants achieved success in TD as opposed to KD (15 versus 5) when they taught with a single demonstration.

The large number of single demonstration instances is an artifact of the experimental design. The skills used in the experiments were chosen to be fairly easy to achieve, there was only a single goal location, and participants were allowed to practice a particular skill before providing an actual demonstration of the skill. This practice opportunity was used more in the TD condition, where people often practiced enough to be able to teach the skill in a single demonstration. To quantify this observation the total movement of the arm during the practice sessions measured in the 7DOF joints space was analyzed. The practice sessions that have less than 10% of average movement of all practice sessions are designated as *minimal practice*. In the KD condition, 17 practise sessions are classified as minimal, while only 4 of TD practise sessions fit this definition This supports the anecdotal observation that practice is more likely to be skipped in the KD condition.

Secondly, as mentioned earlier, participants often do not think of providing keyframes for obstacle avoidance in their first demonstrations. In some cases this does not effect skill success in terms of achieving the goal (*i.e.* partial success) and participants could be satisfied by this since they were not explicitly told to avoid collisions. A large portion of the participants who provided a single demonstration in the KD condition at least achieved

[4] Success levels of skills are treated as ordinal data.

Table 3: Number of participants who achieved different levels of success for goal-oriented skills.

Cond.	# of demo.	Success	Partial Success	Fail
TD	Single	15	4	1
	Multiple	1	5	8
	Total (%)	**16 (46)**	**9 (27)**	**9 (27)**
KD	Single	5	5	1
	Multiple	4	9	10
	Total (%)	**9 (27)**	**14 (41)**	**11 (32)**
KI	Single	4	0	2
	Multiple	6	4	6
	Total (%)	**10 (46)**	**4 (18)**	**8 (36)**
KA	Single	6	2	1
	Multiple	8	4	1
	Total (%)	**14 (64)**	**6 (27)**	**2 (9)**

partial success. This behavior highlights the goal oriented nature of the participants.

Keyframe demonstrations may result in preferable means-oriented skills: Table 4 summarizes the expert ratings for the means-oriented skills taught by participants. Both experts rated the means-oriented skills learned in KD condition higher in all three dimensions on average. The difference was only significant for *closeness to perfection*, and the difference is marginally significant when the three scales are averaged ($Z=2740$, $p=0.06$ on Wilcoxon signed rank test). This distinction is partly related to the difficulty of moving a 7-DOF arm smoothly in the TD condition.

Participants like both TD and KD but bad teachers prefer KD: All ratings were biased towards higher values, and none of the measures showed a statistical difference between TD and KD (based on paired Wilcoxon signed rank tests) for the participant's Likert responses. Participants' ratings are correlated with their success in teaching the goal-oriented skills ($r=.31$, $p<.001$ in Spearman's rank correlation test, assuming Fail:1, Partial:2 and Success:3). As a result, when the participants are grouped into ones that provide a single demonstration and ones that provide multiple demonstrations, we find that

Table 4: Expert ratings of means-oriented skills: Median and Coefficient of Dispersion

Cond.	Expert	*Emphasis*	*Intent*	*Perfection*
TD	1	5.5 (0.27)	5 (0.33)	5 (0.35)
	2	3 (0.29)	3.5 (0.38)	4 (0.3)
KD	1	6 (0.21)	6 (0.17)	6 (0.2)
	2	4 (0.21)	4 (0.24)	5 (0.22)
TD versus KD (Wilcox s.r. test)		Z=2679, p=0.10	Z=2677, p=0.11	Z=2796, p=0.03

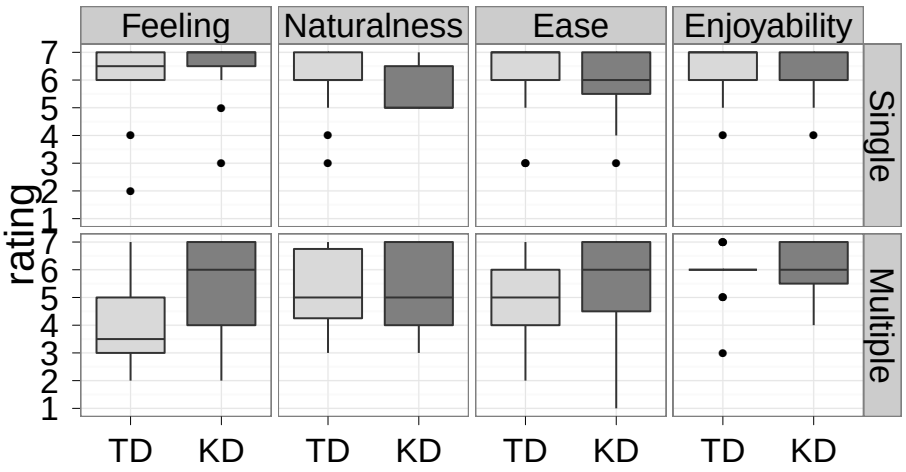


Figure 9: Subjective ratings of TD and KD conditions for goal-oriented skills separated by the number of demonstrations provided by the participant.

participants who provided multiple demonstrations felt more comfortable with keyframe demonstrations (V=98, $p < 0.05$ in unpaired Wilcoxon signed-rank test). This difference is not seen in participants who provided single demonstrations.

Trajectory demonstrations require less time: Providing one demonstration in the TD condition took participants on average 19.34*sec* (SD=7.65), while it took 34.37*sec* (SD=12.79) in the KD condition. There are two reasons for this difference. First one is that participants could freely move the arm before providing a keyframe in the KD condition. Thus, they used more time during the demonstration to think about the next keyframe that they wanted to provide and adjust the arm for it. This is supported by the comparison

of all arm movements in the KD condition between keyframes (which was recorded for reference) and arm movements in the trajectory demonstrations provided in the TD condition. We find that the average arm movement per demonstration per person in the TD condition is about 82% of that of the KD condition, although this difference is not statistically significant (t(112)=1.54, p=.13 on *t*-test). The second reason is that the overhead of the speech commands to record keyframes in the KD condition. Given that the average number of keyframes is 6.71 (see Sec. 4.1.3.3) and assuming giving a record keyframe command takes a second, both reasons seem to be valid.

In the TD condition, since all movements are recorded, participants must constantly progress and cannot pause or adjust as in the KD condition. As mentioned earlier, one manifestation of this was a more thorough practice session prior to TD, as compared to KD.

4.1.3.2 Goal- vs. Means-oriented Skills

Different objective functions for each skill type: As seen in Fig 7, a much larger fraction of participants provide a single demonstration for teaching means-oriented skills, in both TD and KD. Across both conditions, the average number of demonstrations provided for goal-oriented skills (2.37, SD=1.45) is significantly larger than the number of demonstrations provided for means-oriented skills (1.22, SD=0.53) (t(84)=6.18, $p<0.001$ on *t*-test). This highlights a fundamental difference between the skill types: while goal-oriented skills have a well defined objective function, means-oriented skills are subjective and under-specified. Means-oriented skills can vary a lot and were often satisfactory for the participants after a single demonstration.

Open ended questions in the survey reveal more about the difference in the objective functions for the two types of skills. Participants were asked to indicate their criteria of success for each skill that they taught. 15 participants mentioned achieving the goal as their criteria for goal-oriented skills (*e.g.* "The action would most accurately meet its end

goal", "Performing the task correctly", "Touching the point perfectly") while 11 participants mentioned at least one style-related criteria for means-oriented skills. 4 participants mentioned *naturalness* (*e.g.* "more fluid and natural performance", "how naturally Simon emulate the demonstration"), 4 participants mentioned *human-likeness* (*e.g.* "with human characteristics", "seeming less robot-like") and 6 participants mentioned *smoothness* (*e.g.* "how smooth and liquid the motion of the arm is", "more fluid motion", "no choppy movements").

Characteristics of provided keyframes are different for each skill type: The average distance between keyframes in the 7DOF joint space for goal-oriented skills is much smaller (around 47%) than the average distance for means-oriented skills (t(38)=-3.94, $p<.001$ on *t*-test). It is hypothesized that participants are providing different types of keyframes within a single demonstration. For goal-oriented skills we see a distinction between keyframes that are instrumental to the goal of the skill, and the keyframes that lets the robot avoid obstacles. Similarly in means-oriented skills we see a distinction between keyframes that actually give the skill its meaning and make it recognizable and keyframes that are waypoints. Participants provide a large number of frames that are close to one another around the goal of goal-oriented skills. For means-oriented skills, they provide less frames that are separated by a larger distance. For both types of skills the waypoint keyframes or obstacle avoidance keyframes tend to be further apart. A statistical difference in the average number of keyframes for goal-oriented skills (6.75, SD=1.89) and means-oriented skills (6.21, SD=2.17) is not observed (t(65)=1.11, p=.27 on *t*-test).

4.1.3.3 Keyframe demonstrations vs. Iterations

A larger fraction of the participants achieve success in the KI condition as compared to the KD condition (Table 3). Since the order of these two conditions was not counterbalanced, this difference partially involves the improvement that comes with more experience in teaching. However these results show that the iteration process was effectively

used by the participants, despite the increased number of commands and the complex interaction cycle. Note that 6 out of the 22 participants in the KI condition did not use the iteration process, *i.e.* they were satisfied the skill performance after the initial demonstrations, which is provided in exactly the same way as in the KD condition (Fig. 10). From the participants who used the iteration process, 10 participants provided a single iteration, however the iteration often involved several editing commands.

An interesting observation is that the number of keyframes in the demonstrations given by a participant varies less in the KI condition. The average number of keyframes provided within a demonstration for each participant is not very different in the KI (7.62, SD=1.48) and KD (6.71, SD=1.91) conditions. However the standard deviation in the number of provided keyframes across demonstrations of a participant is larger in the KD condition (1.26, SD=0.94) as compared to the KI condition (0.46, SD=0.58). By starting from the previous learned skill, the iteration process limits the deviation in the number of keyframes in the provided demonstrations which can be an advantage while learning from keyframe demonstrations if the initial demonstration is relatively good.

4.1.3.4 Effects of the Starting Skill for Keyframe Iterations

The fraction of participants who achieve success is largest for the KA condition with 64% among all the others (TD: 46%, KD: 27%, KI: 46%, Table 3). As in the KI condition, experience in teaching the robot might be contributing to this improvement. However, this result indicates that an iterative process starting from a rough, often failing skill is potentially the best option in terms of achieving successful goal-oriented skills.

The survey involved a question asking whether having a rough skill to start from in the KA condition made it easier or harder, or whether it did not matter in comparison with providing the initial demonstration themselves as in the KI condition. 12 participants responded that it made it easier, while 5 said “harder” and 5 said “did not matter”.

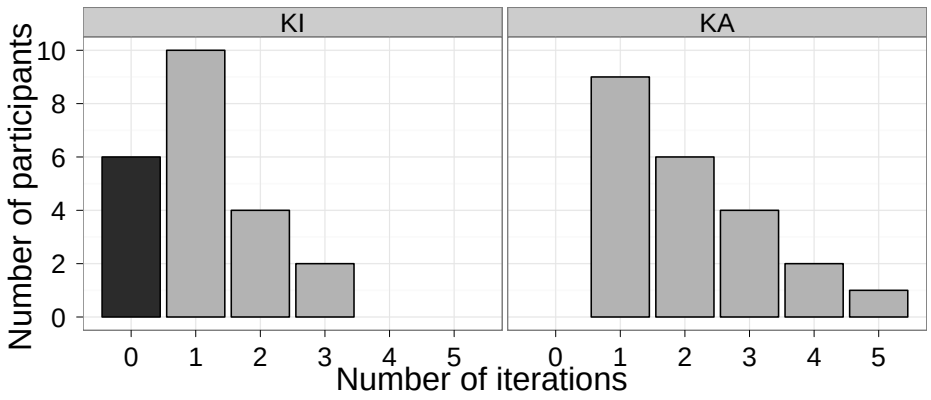


Figure 10: Histogram of number of iterations provided by participants in KI and KA conditions. For the KI condition "0" indicates the participants who only provided an initial demonstration and did not provide any iterations.

4.1.4 Implications of Experiment I Results

4.1.4.1 Benefits of each Demonstration Method

The results of our experiment show that both trajectory and keyframe demonstrations are viable methods of interaction for kinesthetic teaching in LfD. Each has advantages and users are positive towards both of them.

Trajectory demonstrations are clearly intuitive for a naïve teacher, and there is the benefit that many existing LfD methods are designed for learning skills from trajectory data. Trajectories allow complicated skills to be taught, and are particularly appropriate when dynamics (*e.g.* speed) information is a key component. However, it might be hard for teachers to manipulate a high-degree of freedom robot or sustain smooth trajectories over the course of a demonstration. In the experiment, this resulted in longer practice sessions for trajectory demonstrations, as well as means-oriented skills that achieved lower expert ratings.

Keyframe demonstrations are robust to these noisy and unintended motions during a demonstration. Their sparse nature result in a modular representation, which may be useful in generalizing a skill to new situations. For example, existing motion planning methods can easily be used to navigate between keyframes to execute salient aspects of the

skill while avoiding obstacles. Additionally, it may be easier to deal with time alignment between multiple demonstrations.Furthermore, extensions like keyframe iterations are relatively straight forward to implement.

A drawback of keyframes is the lack of timing information. It was observed that some participants tried to achieve slower movements or stops by providing a large number of very close or overlapping keyframes. Several participants mentioned wanting speed related commands.

4.1.4.2 Skill Types and Demonstration Methods

The experiment reveals the different nature of goal-oriented and means-oriented skills. The former is defined by *success* while the latter by *style*. Moreover, in the goal-oriented skills, only a portion of the skill's motion contributes to success whereas in means-oriented skills the entire motion contributes to the style.

People gave more demonstrations for goal-oriented skills. Since means-oriented skills can vary a lot, they were often satisfactory after a single demonstration. This was particularly true for keyframe demonstrations, since some users had a hard time manipulating the robot's arm, especially during the start of a skill. In goal-oriented skills, this usually did not impact task success (e.g. initial motion of the arm was not that important for touching a point). However, this does have an impact when style is the objective. Thus, users often needed to correct the style by giving multiple demonstrations for the means-oriented skills in trajectory mode.

For goal-oriented skills, participants often gave multiple demonstrations due to the lack of fine control with keyframes, most notable being the timing (velocity) information. In addition, some participants did not provide keyframes for obstacle avoidance in their first demonstrations. The robot often did not perform a skill as intended after the first demonstration in keyframe interactions, prompting users to improve the skill with more demonstrations.

It was also observed that skill types have an effect of types of keyframes that are provided by the user. Waypoint keyframes are common in both of the skill types. Goal keyframes (keyframes that are closer together near a goal) and style keyframes (keyframes that are placed strategically to do the gesture, usually further apart) can clearly be seen respectively for goal-oriented and means-oriented skills.

4.1.4.3 Designing Keyframe Interactions

The experiment explored different interaction mechanisms for a keyframe approach. Since keyframes temporally segment the demonstration, it is easy to apply an iterative interaction mechanism, and the experiment showed that people were able to use this to achieve greater skill success. Additionally, in an iterative interaction, people do not stray too far from their initial demonstration, thus emphasizing the importance of the starting skill. The experiment showed that people were even able to use the iterative process to adapt a starting skill that was not their own, and many said that this made the teaching process easier.

As mentioned above, all keyframes are not equal, people think about them in different ways (*e.g.* goal frames, via points, *etc.*). The distinction between these types of keyframes is important information for the underlying learning algorithm that human partners can easily provide.

4.2 Hybrid Demonstrations

The findings of Sec. 4.1 suggest that both keyframe and trajectory demonstrations have their own benefits. Based on this insight, this section introduces *hybrid demonstrations* (HD) to take advantage of the complementary nature of trajectory and keyframe demonstrations.

The ability to provide both keyframe and trajectory information in the context of a single demonstration will be useful and intuitive for a variety of skills and even combination of skills *e.g.* scooping and then serving). Keyframe demonstrations (KD) allow the teacher to freely manipulate the robot and carefully configure it before recording the keyframes of the demonstration. Unlike trajectory demonstrations, this allows collecting demonstrations

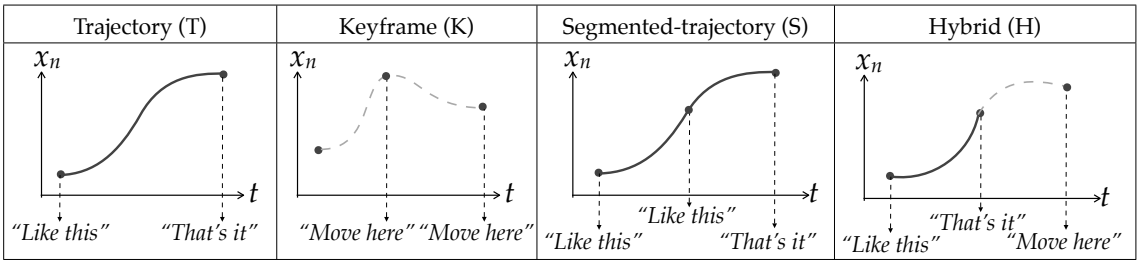


Figure 11: The possible interaction flows of the hybrid mode. The dots correspond to start/end points or keyframes, the solid lines to user demonstrated trajectories and the dashed lines to splines between keyframes.

free of movement noise and mistakes. On the other hand, demonstrating complex curved movements requires a large number of keyframes when using keyframe demonstrations and would better be served with trajectory demonstrations (TD). Hence, this section introduces a new interface for LfD which merges trajectory and keyframe demonstrations in a single interaction. This *hybrid* interaction scheme allows the teacher to give both keyframes and trajectory segments in their demonstration (see in Figure 11). During a demonstration, the teacher can provide a keyframe by moving the arm to a desired position. At any point, the teacher can provide a trajectory demonstration. The teacher can combine these in any order resulting in four different kinds of demonstration: pure keyframe, single trajectory, segmented trajectory, and hybrid demonstrations.

The hybrid demonstrations will give teachers more tools at their disposal to program robots in ways they find intuitive. This approach was demonstrated at the AAAI 2011 LfD challenge [5], on the PR2 robot, where anecdotal evidence shows that this hybrid-mode was intuitive for conference goers. This motivated the development of a method to learn from hybrid demonstrations. Chapter 5 introduces a method to learn from hybrid demonstrations. In this chapter, an early implementation will be used to test hybrid demonstrations with teleoperation.

4.3 Experiment II: Improving Teleoperation for LfD

This thesis concentrates on *kinesthetic teaching* since it is intuitive and easy to use. It does not suffer from the *correspondence problem* between the teacher and the robot. The resulting demonstrations are restricted to the kinematic limits (e.g. workspace, joint limits) of the robot. Moreover, extra hardware/instrumentation, such as motion capture devices, is not necessary.

This thesis concentrates on *kinesthetic teaching* since it is intuitive and easy to use. It does not suffer from the *correspondence problem* between the teacher and the robot. The resulting demonstrations are restricted to the kinematic limits (e.g. workspace, joint limits) of the robot. Moreover, extra hardware/instrumentation, such as motion capture devices, is not necessary.

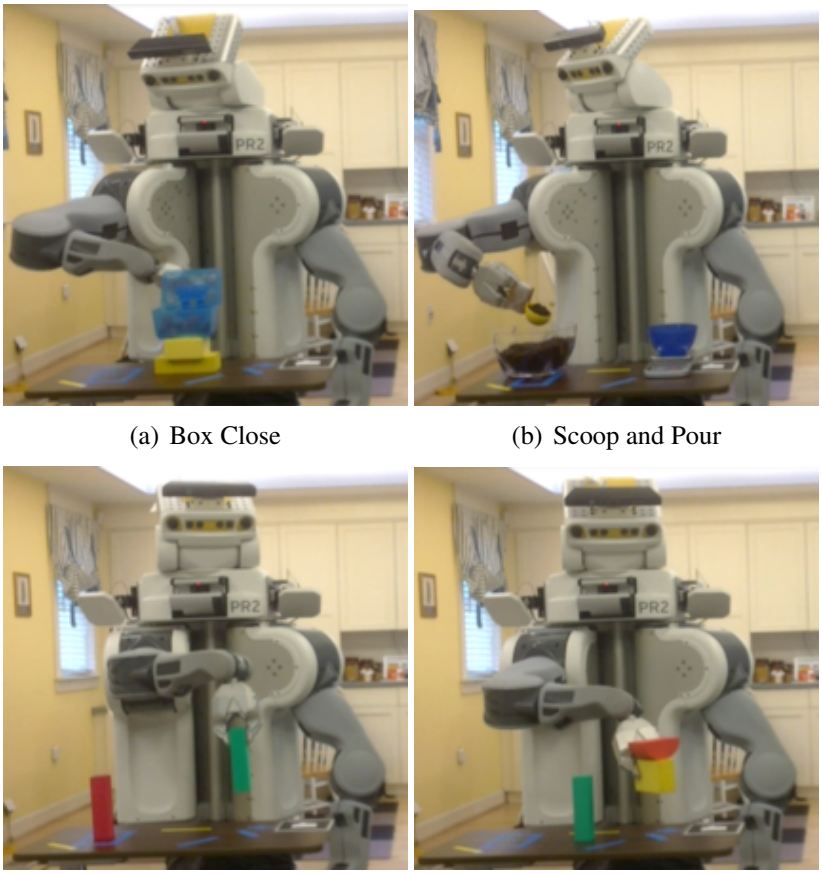
Kinesthetic teaching may not be always available. It requires that the robot and the teacher be co-located and that the teacher can manipulate the robot. This might not be possible if the robot is distant (*e.g.* a robot on the moon), the robot or the environment is dangerous (*e.g.* a disaster area) or the scale of the robot does not permit it (*e.g.* endoscopic surgery). This is when *teleoperation* becomes important. It does not suffer from the correspondence problem but it is difficult and requires a teleoperation device. Based on these motivations, this section aims to improve teaching interactions for teleoperation.

The experiments presented in this chapter follow the protocol described in Sec. 3.2.

4.3.1 Skills

This experiment has a total of four main tasks for participants to teach the robot, shown in Fig. 12, all of which were designed such that they are achievable with all the interaction modalities and demonstration strategies. The tasks involve the use of a single arm of the robot.

- Box Close: The goal of this skill is to move the robot arm such that it closes the lid of an open box.



(a) Box Close (b) Scoop and Pour

(c) Stacking (d) Cup and Saucer

Figure 12: Tasks used in our experiments

- Scoop/Pour: A spoon is placed in the robot's gripper and the goal is to transfer as many coffee beans as possible from a big bowl to a nearby smaller bowl.

- Stacking: The goal of this skill is to move the robot arm to grip a relatively slim block with a square cross-section and then place it on top of another similar block.

- Cup/Saucer: A hemispherical block is placed on another relatively thin rectangular block from its circular side. The top block falls if the arm moves too fast or the orientation deviates. The aim is to transfer these blocks into a rectangular region by avoiding an obstacle.

There are also two practice skills to help familiarize the participant with the abilities of the robot. One is called "Orient and Place". In this skill, the robot holds an oblong prism and the goal is to make this fit within a gap of two blocks placed on the table. The gap is placed such that the participant needs to both manipulate the position and orientation of the robot's end-effector. The other practice task is "Peg in Hole". In this task, a vertical slim block should be grasped, inserted through a horizontal hole, and then be placed back near its original position.

4.3.2 Pilot Study: Kinesthetic Teaching versus Teleoperation

The pilot study compares Kinesthetic Teaching (KT) and Teleoperation (TO) in an LfD setting with naïve teachers. The teachers are instructed to teach the PR2 robot a set of skills in both conditions. The aim is to look into the characteristics of these two modalities and highlight the participant's comfort and the robot's skill accuracy in using these.

For the purposes of this pilot study, only trajectory demonstrations are used. The off-the shelf GMM+GMR method, [21], is used to learn these trajectories. The state space is the end-effector pose of the robot.

To compare the KT and TO input modalities, a within-subjects experiment is used. Every participant taught two skills, Box-Close and Scoop/Pour, to the robot in each of the

modalities. The study had 9 participants, 5 females and 4 males, all of whom were Georgia Institute of Technology students. Their ages were between 23 and 32 with a median of 25. None of the participants had any previous machine learning and robotics experience.

4.3.2.1 Research Questions and Metrics

The following research questions will be addressed with this pilot study:

Q1 Do naïve teachers prefer kinesthetic teaching over teleoperation?

Q2 Which input modality results in a more successful skill model?

To answer these questions, the participants were asked to rate the *ease of use*, *enjoyability* and *accuracy* of the method and the extent to which they thought they would *improve at using the modality, given time* with a set of 7-point Likert-scale survey questions. An open-ended question was also asked to get the overall impression from the participant. The question was phrased as "If you bought this robot to use at your house, which modality would you prefer and why?". In addition to the survey, the input modalities KT and TO are compared with respect to the skill-oriented metrics: duration of demonstrations; and success of the learned model.

4.3.2.2 Survey Results

The Wilcoxon signed rank test is used to evaluate the survey (see Figure 13). A summary of the results obtained is given below.

Kinesthetic teaching was rated easier: The median answer to the ease-of-use-of-modality question was 6 for the KT case, whereas it was 5 for the TO case.The answers are significantly different from one another ($p = 0.05$). This result was expected due to the fact that people are more accustomed to a kinesthetic type of teaching, *i.e.* it occurs naturally in human-human interactions. Moreover, with this interaction method, the users have more control over robot's joints, can more easily adjust their perspective to see more of the workspace and be more *situated*.

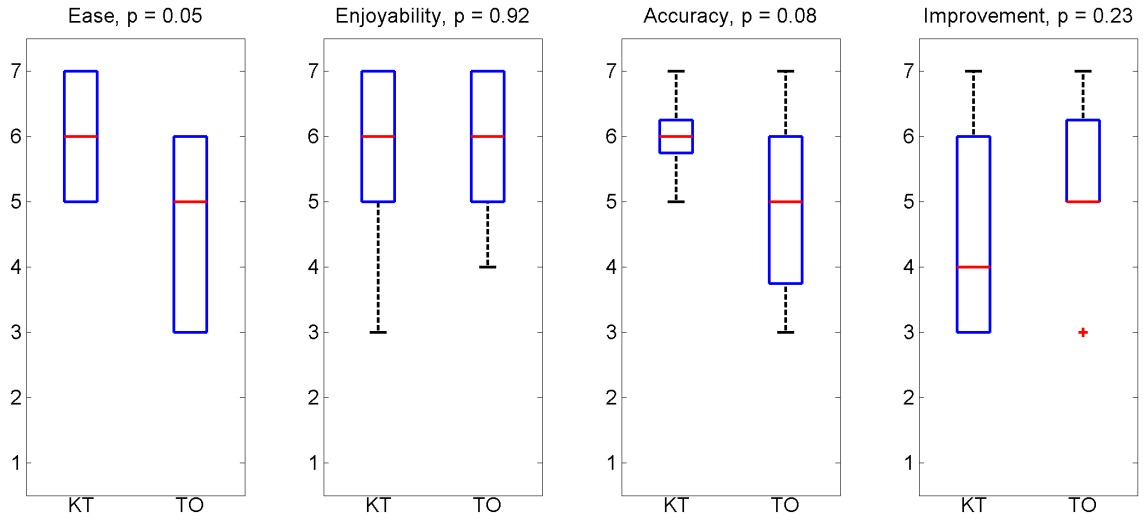


Figure 13: Box and whisker plots of survey replies for the pilot study of Experiment II.

Participants enjoyed both methods: Both methods were rated highly on the enjoyability scale with no significant difference between them.

Participants tend to think that they can give more accurate demonstrations with the kinesthetic teaching method: Although this is not significant, there is a a trend ($p = 0.077$).

Majority preferred kinesthetic: According to the open-ended question responses, a majority of participants (7 out of 9) preferred KT over TO, with 6 participants citing their reason being its "ease" of use.

4.3.2.3 Skill Results

In addition to the survey results the study looks at skill-specific success rates and demonstration durations. The end-state of the box in the Box Close skill (open or closed) and the amount of coffee beans transferred for Scoop/Pour are defined as the success metrics.

The Box Close skill was completed successfully by all participants but one using both modalities. In the Scoop/Pour skill demonstrations, participants transferred more coffee beans with KT than TO ($p < 0.05$ in paired t-test). This is not always reflected in the learned tasks. There are two probable causes for this. First, participants may provide subtle

but useful assistance (e.g. rocking the spoon) during kinesthetic teaching since they are more accustomed to this form of interaction. However, these are smoothed out by learning. Second, the distribution of the coffee beans before executing the skill was not controlled [5]. After a demonstration, a dent is left in the distribution and the learned skill will try to scoop from around the demonstrated region but will not get as many coffee beans due to the dent.

The participants were faster at providing demonstrations with KT for Scoop/Pour ($p < 0.05$) than TO. For Box Close, people were faster on average but not significantly ($p = 0.09$). This is partly due to 2 outlier users who took some time to realize they needed to move some of the robot joints (shoulder joints) that were away from the end effector in KT modality. Overall KT leads to more successful demonstrations in a shorter amount of time.

4.3.3 Experiment: LfD with Teleoperation

The results of the previous pilot study showed that there is a gap between kinesthetic teaching and teleoperation in terms of usability in an LfD setting, with kinesthetic being easier to use and leading to more successful demonstrations. However, as motivated before, teleoperation is applicable in certain scenarios where kinesthetic teaching is not. Thus, this experiment looks at novel demonstration strategies aimed at improving a teleoperation teaching interaction. The *trajectory demonstrations* (TR) and the previously introduced *keyframe demonstrations* (KF) and *hybrid demonstrations* (HY) are tested for teleoperation.

4.3.3.1 Research Questions

The following research questions will be addressed by this experiment:

Q1 What is the naïve teacher preference amongst trajectories, keyframes and hybrid demonstrations?

Q2 Does the addition of keyframe and hybrid demonstrations for LfD with teleoperation result in better demonstrations?

[5] This was done to keep the experiment going without interruption, but in hindsight was a bad decision.

It is hypothesized that the new strategies will enhance the user interaction with the teleoperation device both in terms of "ease of use" as well as providing better demonstrations. This experiment is setup to first compare the individual utility of keyframes and trajectory strategies and then compare them both against the hybrid strategy. The three demonstration strategies are compared based on survey results (Likert scale questions and open-ended responses) as well as characterizations of the demonstrations data provided with the different strategies. The skill success metrics was not the focus of this experiment.

4.3.3.2 Experiment Details

The PR2 robot is used as in the pilot study. The state space of learning was again end-effector poses. The learning of keyframes and trajectories are done as described in Sec. 4.1.1 based on GMMs. In hybrid demonstrations (HY), the user is allowed to give both keyframes and trajectory segments in their task demonstrations as described in Sec. 4.2 and illustrated in Fig. 11. Each segment in a hybrid demonstration is learned separately. If there are multiple demonstrations, the first step for learning is to match segments. This is done by treating starting and end points of portions as keyframes, and then do keyframe learning on these. The portions are then matched according to the clusters that their start/end points belong to. For generating motions each segment is again treated separately and the resulting trajectories are merged together. This is not an ideal way of learning from hybrid demonstrations but it was enough for the purposes of this experiment. A more comprehensive way of learning is described in Chapter 5.

A within-subjects study where every participant did all the 3 strategies and performed all 4 skills mentioned in Sec. 4.3.1 was performed. The Box Close skill was made harder by requiring participants to make the lid "click" (by pushing it down) after closing it. There were 12 participants, all male, from the campus community (different from the ones who participated in the pilot). Their ages were between 18 and 47 with a median of 21.5. Only one user was a first year Ph.D. student in the Robotics program. The others were not experts

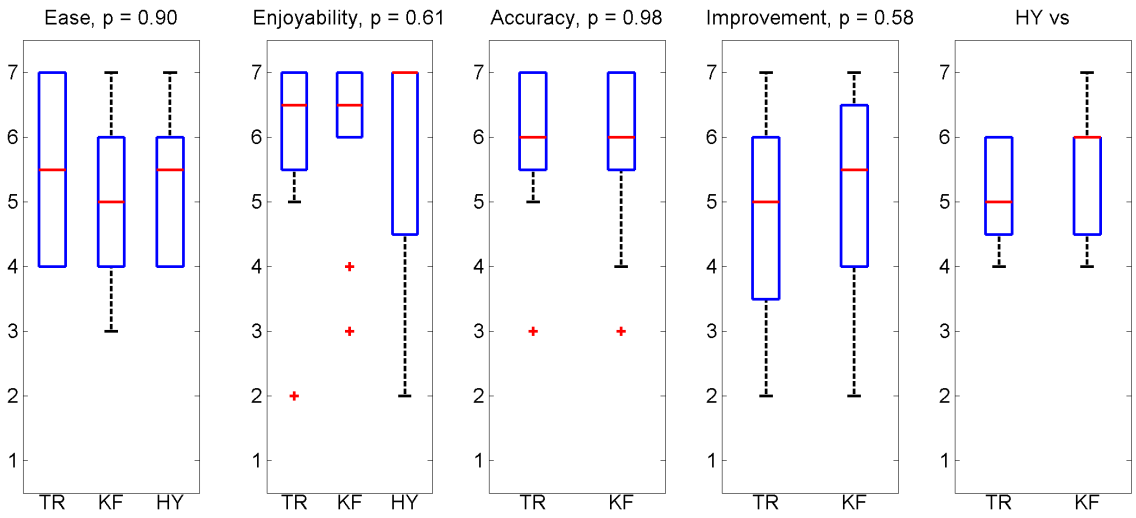


Figure 14: Results for choice questions on the survey for the experiment. The p-values are obtained with the Friedman's test when comparing all methods and the Wilcoxon signed rank test when comparing just TR and KF.

in any related field and none of them had used a teleoperation device before.

Some skills can be more efficiently solved using specific or a combination of strategies. For example, the stacking task can be better suited for demonstrations using the keyframe strategy as it requires only a set of linear translations, whereas the Cup/Saucer task requires the use of trajectories as they provide control over the speed of the arm. Without speed control, the hemispherical block has more tendency to fall down.

Each user demonstrates 2 skills per strategy. The skills differ across TR and KF. Then one task from TR and one task from KF is chosen for HY (e.g. (TO: T_1 T_2)$\rightarrow$(KF: T_3 T_4)$\rightarrow$(HY: T_1 T_4) where T_x denotes one of the four experimental tasks). The study partially counterbalanced the strategy and the skill order. Half of the experiments started with TR and the other half with KF. Note that there were $(3 \times 2) \times 12 = 72$ interactions which are distributed evenly among the related conditions (e.g. 24 per demo, 18 per task, 6 per demo and skill combination).

4.3.3.3 Survey Results for Keyframe, Trajectory, and Hybrid

The Fig. 14, presents the results of the survey questions.[6] None of the replies are statistically significant between the strategies, so no differential conclusions can be drawn. There was positive bias in people's answers across all the strategies. For example, all of them were rated enjoyable, with medians being close to the upper limit. This is in part due to the novelty effect of interacting with a robot, but the positive bias also indicates that our interaction strategies were acceptable to the participants.

Participants subjectively reported that all of the interactions were easy. However, this was not the observation during the experiment. It is difficult to manipulate a robot with a teleoperation device, and people clearly struggled at times. Nevertheless, the perceived ease is a positive for teleoperation and the interaction methods and shows that the participants were comfortable with the design and use of these strategies.

Users also thought that the methods were accurate. This is interesting since the keyframe method does not seem intuitive at first, but it received very similar perceived accuracy ratings compared with the more intuitive trajectory method. The improvement results indicate that the users think that they could do better with more experience, which is especially true for such a teleoperation scenario.

4.3.3.4 Open-ended Responses on Keyframe vs. Trajectory

In an open-ended response question, participants were asked to directly compare keyframes and trajectories. In their responses, 9 out of 12 users preferred keyframes over the trajectories. Six of the participants who chose keyframes mentioned giving more "efficient" demonstrations and "not recording any mistakes". Two of the users admitted that they were not very proficient with the teleoperation device and felt more comfortable with the keyframe mode. All three users who chose trajectory mode complained about "having to

[6] Only two of the questions were asked for HY. This was to shorten the survey to minimize fatigue. In addition, the HY condition it is biased since it was not counterbalanced; people inherently improved and became more accurate by the time they completed this.

Table 5: Mean (and standard deviation) of demonstration duration and distance.

	Trajectory	**Keyframes**	**Hybrid**
Duration(seconds)	50.69 (26.26)	72.45 (30.36)	59.84 (31.13)
Distance(meters)	3.65 (1.46)	2.12 (0.26)	3.08 (1.3)

give many poses" with the keyframe strategy; showing some concern for the loss of information with keyframes.

4.3.3.5 Analysis of Keyframe vs. Trajectory Demonstrations

The average number of keyframes per task was 10.25 ($SD = 3.77$). Table 5 shows the mean and the standard deviation of distance covered and the average time taken to complete a task in each of the modes. There seems to be an inverse relationship between the time taken and the distance covered. There is a significant difference for the demonstration duration ($t(23) = -2.67, p = 0.014$) and a significant difference for distance traveled by the robot end-effector between trajectories and keyframes ($t(23) = 4.80, p < 10^{-4}$). The latter result is due to the fact that the robot moves nearly in a straight line between keyframes but trajectories include the unnecessary motions of the user. These results indicate that the participants spent more time positioning the arm. This in turn resulted in a good selection of keyframes as the arm completed the task by traversing a smaller distance, making it more efficient.

The accuracy of the trajectories as perceived by the participants and as obtained by the quantitative measures can be misleading as the participants were more interested in task completion rather than providing clean and noise free demonstrations. The trajectories had a lot of hand jitter and unnecessary motions that would be very hard to learn from. However, on reviewing the demonstrations obtained in the keyframe mode, they were noise free (i.e. little or no unnecessary keyframes) which is much better suited for input to a learning algorithm. This attribute is highlighted in Fig. 15, showing an example keyframe and trajectory demonstration of the Box Close task.

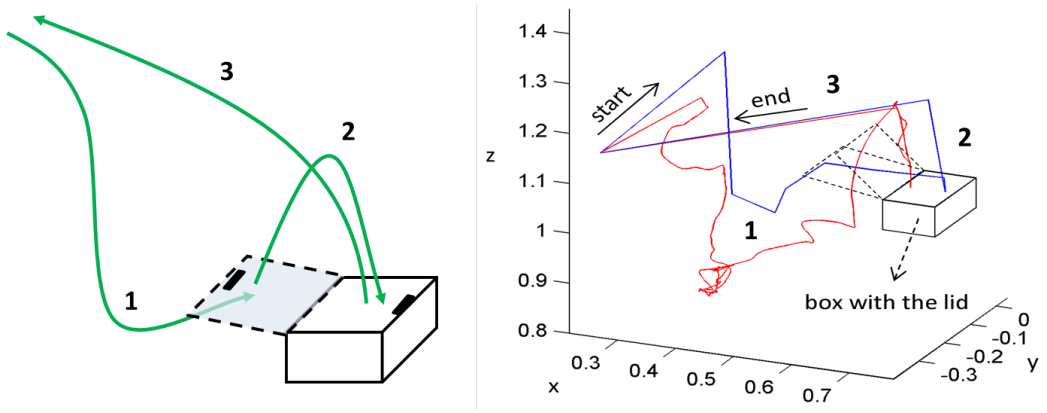


Figure 15: Comparison of Trajectory (Red) and Keyframe Demonstrations (Blue). Note that the trajectory is highly noisy. The left image shows a desirable trajectory for closing the box lid.

Additionally some tasks were hard to perform using the keyframe mode. For example, the Scoop/Pour task and the Cup/Saucer required fine control as well as speed control of the arm. We can therefore say that the keyframe mode was not sufficient to solve all the tasks efficiently.

4.3.3.6 Survey and Open-ended Responses on Hybrid mode

On comparing hybrid demonstrations with the other two techniques, the results were encouraging. Fig. 14 shows that hybrid is rated easy and enjoyable. People not only thought it to be a valuable addition to the interaction modes, many participants were able to figure out efficient ways to combine keyframes and trajectories. The last column of Fig. 14 is people's response to questions asking them to rate how much they prefer the HY method over the TR and KF. People were positive towards the hybrid mode with a median of 5 for HY vs TR and median of 6 for HY vs KF and all users were at least neutral (4) towards HY.

The second open choice survey question was designed to compare the hybrid mode with the other two modes and provide reasons for their choices. 11 of the participants thought hybrid was a valuable addition and they preferred it over keyframes and trajectory

modes. We would like to highlight two characteristics mentioned by the participants in the survey question. 6 of the participants preferred the Hybrid mode due to the efficiency of the interaction and 5 of the participants highlighted the ability for precise control. Specifically several mentioned how it is easier to demonstrate gross motions using keyframes and fine motions using trajectories. One user mentioned "a combination keyframes and trajectories" would be a valuable addition before being informed about the hybrid strategy.

4.3.3.7 Analysis of Hybrid mode Demonstrations

In the final analysis of the hybrid strategy, some of the choices the participants made are highlighted, specifically how they choose keyframes and trajectories depending on the type of skill. It was observed that the keyframe mode was primarily used for gross motions from location A to B, for linear motions or when only the end point mattered. The trajectory mode was primarily used when the task required non-linear motions or fine control over the speed. An example scoop and pour demonstration can be seen in Fig. 16. It can be seen that scooping and pouring is done with trajectories and going from one bowl to the other with keyframes.

We analyze the choices of the users in the hybrid mode for specific tasks.

- In the Cup/Saucer task, 5 out of 6 participants that did this task with hybrid used the trajectory mode to move the cup because it gave them more control over the speed.
- In the Scoop/Pour task, $5/6$ used trajectory for scooping, $2/6$ for transferring, and $5/6$ for pouring.
- In the Close the Lid task, $3/6$ users moved under the lid with the keyframe method and all of them used trajectory mode to close the lid. 1 of the users then used the keyframe method to push the lid to its place.
- In the Stack the Block task, 4 people used keyframes to move to the first block, 2 to go to the next and 3 to stack. Among the users, one of them did this task with only

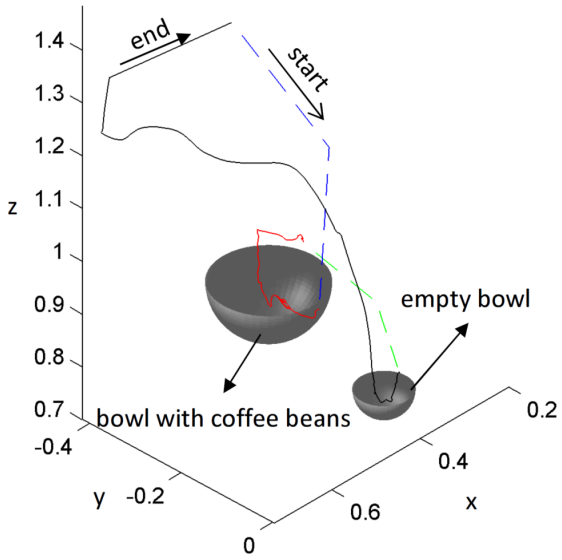


Figure 16: An example hybrid demonstration for the scoop and pour task. Dashed lines represent keyframe portions and continuous lines represent trajectory portions. Different colors correspond to different demonstration segments.

keyframes, which is arguably the best option.

In general, people tried to take advantage of keyframes and trajectories wherever appropriate. Participants show a trend of choosing trajectory for fine control and keyframes for gross point-to-point motions. With more practice, users can develop even better strategies to more efficiently achieve the skills with the hybrid strategy.

4.3.4 Discussion of the Teleoperation Experiments

The pilot study showed that users preferred kinesthetic teaching over teleoperation as it is more intuitive and more situated. They were still positive towards teleoperation. The users did not have any previous experience with the PR2 robot nor have they ever had any experience with a teleoperation device. This makes the already steep learning curve of teleoperation even steeper.

The introduction of keyframes and the hybrid strategy made the LfD interaction with teleoperation more suitable. Participants quickly figured out the concept of keyframes and learned how and when to provide them. It took users a couple iterations of looking at the

robot replay their demonstrations at most to understand the steps necessary to correct the position of the keyframes. It can be seen that the time taken for providing keyframes was greater than trajectories, shown in Table 5. This is attributed to two reasons; one, the users spent time to think where the poses must be given and to position the robot accurately and two, they spent time giving the speech command and waiting for the robot to confirm. This in fact supported the hypothesis that users were ready to spend that extra time in providing keyframes because the robot demonstrations were less prone to noise.

Furthermore some participants, with continued interactions, were able to gain insight into the properties of keyframes as envisioned by the experimenters. Specifically, they were able to understand that keyframes assume constant speed between them and therefore do not encode any velocity related information. Two participants specifically mentioned that "keyframes are not good when speed control is required". This only goes to show how naive users using a few interactions were able to grasp the details of the interaction strategies.

Given these characteristics of the participants in the study, it is necessary to highlight an aspect that was common to most of the users. The results indicate that the users concentrated more on task completion rather than providing good demonstrations, although they were encouraged to give smooth demonstrations. They perceived the robot being accurate during the replays, however their trajectories often contained noisy, unnecessary and imprecise portions which makes learning difficult.

4.4 Summary

These experiments compared different methods of interaction for LfD with everyday people totalling 55 participants. The first experiment, Experiment I, focused on the effects of different types of demonstrations for kinesthetic teaching, and showed that trajectory and keyframe demonstrations have their relative advantages. It also explored different interaction schemes that a keyframe representation makes possible (iterations and adaption)

and showed their success with human teachers. Based on the results, a hybrid demonstration approach was introduced. It was also observed that the users did not mind providing demonstrations with noisy and unintended motions as long as the skill was successful, hinting at goal oriented behavior.

The second experiment started by comparing teleoperation and kinesthetic teaching with a pilot study, finding the expected result that kinesthetic teaching is more intuitive and easier to use. Then, a follow-up experiment that applies keyframes and hybrid demonstrations to teleoperation was performed. It was shown that naïve users can effectively take advantage of hybrid demonstrations for LfD and prefer this mode of teaching. In addition, it was observed that the teachers did not care too much about how they demonstrate as long as they demonstrate a successful instance of the skill. This is the same goal oriented behavior that was observed in the first experiment.

There are two main take away points from these experiments for the purposes of this thesis. The first one is that keyframes are a viable input method for LfD. They are robust to noisy, inconsistent and unintended demonstrations. In addition, keyframe demonstrations are easy to modify. Experiment I described keyframe iterations that take advantage of this by letting the user edit parts of the existing keyframe skills. The users preferred to use keyframes more as the demonstration task got harder, *e.g.* when they had trouble moving the robot arm in kinesthetic teaching or teleoperation. Moreover, it was shown that naïve users were able to use hybrid demonstrations effectively when teaching skills. This led to the development of a framework that can learn from trajectory, keyframe and hybrid demonstrations as described in Chapter 5. The second one is that people are **goal oriented** in their demonstrations as observed in both of the experiments. These points are highly related in the sense that keyframes help the teacher highlight salient parts or subgoals of the skill which will be used to learn goal models as described in Chapter 6.

CHAPTER V

KEYFRAME BASED LEARNING FROM DEMONSTRATION

The findings of Chapter 4 suggest that both keyframe and trajectory demonstrations have their own benefits. Based on this insight, *hybrid demonstrations* (HD) was introduced to take advantage of the complementary nature of trajectory and keyframe demonstrations as illustrated in Fig. 11.

The experiment described in Sec. 4.1 evaluated keyframe demonstrations against trajectory demonstrations from an HRI perspective, revealing a set of advantages and disadvantages for each. Trajectory demonstrations were more intuitive for naive users, and allowed teaching complex skills where speed information is important. However, it was hard for users to move a high dimensional robot arm smoothly, requiring more practice and often resulting in noisy and undesirable movements. Keyframe demonstrations, on the other hand, were not affected by unintended, noisy motions. A drawback of keyframe demonstrations is the lack of timing and speed information for keyframe poses.

Keyframe demonstrations allow the teacher to freely manipulate the robot and carefully configure it before recording the keyframes of the demonstration. Unlike trajectory demonstrations, this allows collecting demonstrations free of movement noise and mistakes. On the other hand, demonstrating complex curved movements requires a large number of keyframes when using keyframe demonstrations.The proposed hybrid demonstrations can have both trajectory or keyframe segments to combine the advantages of both types of demonstrations.

This chapter develops a framework which can learn from trajectory, keyframe and hybrid demonstrations in a unified way. The method converts all demonstrations into

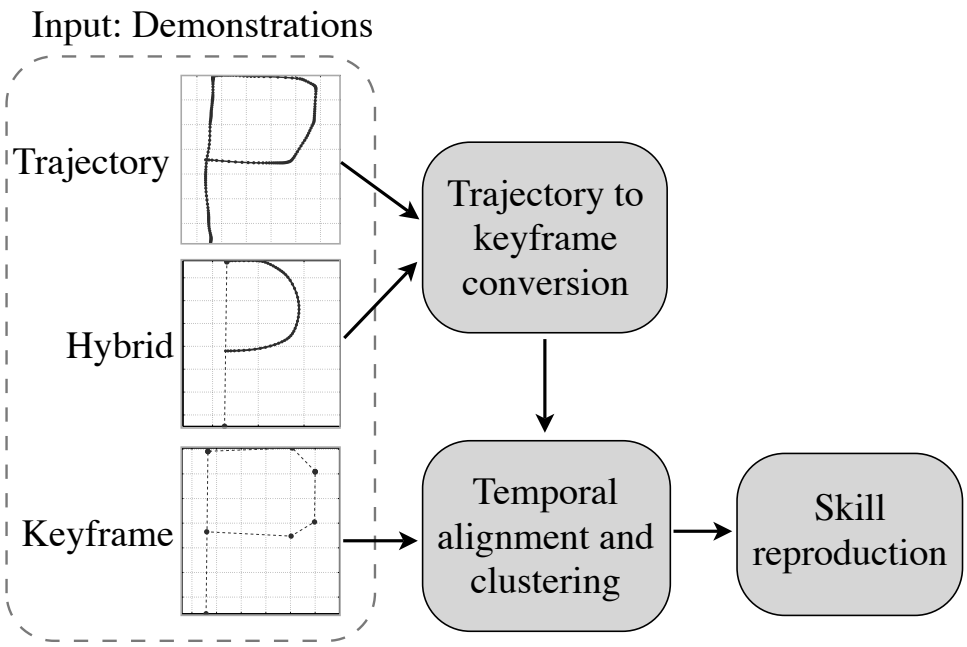


Figure 17: Overview of the steps involved in KLfD.

keyframes and produces a skill model. The entire learning approach is called as Keyframe-based LfD (KLfD). This fits within the *Algorithm* box of the system in Fig. 5. The work presented in this chapter is published in [3].

5.1 Details of the KLfD Framework

This section describes the KLfD implementation details. An overview of the steps involved in KLfD is given in Fig. 17. For illustrative purposes, this chapter uses 2D data for the capital letter P throughout, as seen Fig. 18. Details on how the data is generated is given later in Sec. 5.2.1.

5.1.1 Trajectory to Keyframe Conversion

The method supports input of trajectory, keyframe, or hybrid demonstrations. For trajectory and hybrid demonstrations, a preprocessing step is added to convert trajectory segments into keyframe sequences. To do so, the Forward-Inverse Relaxation Model (FIRM) [72] is used.

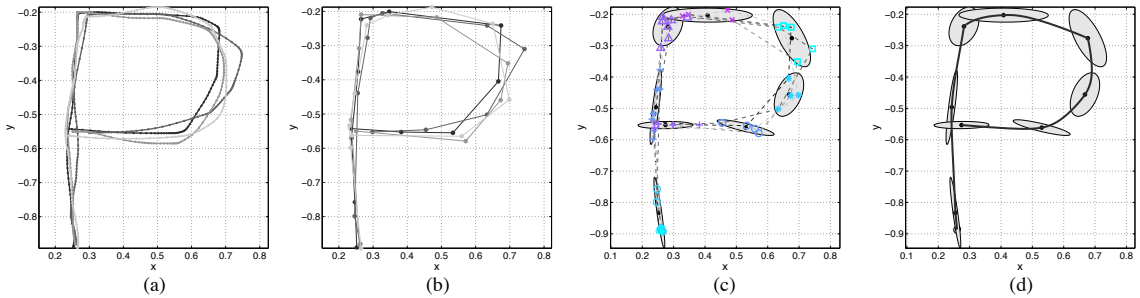


Figure 18: Illustration of the steps in learning with keyframes using a 2D example. (a) Four demonstrations of the letter P given as continuous trajectories in 2D (b) Data converted to keyframes (c) Clustering of keyframes and the reulting model (d) Trajectory produced from the learned model.

The trajectory to keyframe conversion starts with treating the end-points of the trajectory segments as keyframes. Then a fifth order spline is generated by utilizing the positions, velocity and acceleration information at the keyframes. If velocity and acceleration data is unavailable from the demonstration itself, the smoothed first and second derivatives of the trajectory is used. Note that Using velocity and acceleration data along with position data helps to keep some of the dynamics of the demonstration. The original trajectory and the generated trajectory are compared to locate the point which has the largest Euclidean discrepancy at any given time. This is in essence leveraging the skill reproduction method described later in Sec. 5.1.3, which can be seen as an extension of Lowe's method [49]. Fig. 18(b) shows keyframes obtained from the four trajectory demonstrations of the letter P shown in Fig. 18(a).

5.1.2 Aligning and Clustering

The purpose of thisaligning and clustering is to come up with an action model given multiple keyframe sequences [1]. Given several demonstrations of a skill, one common problem for LfD techniques is to temporally align the demonstrations before using them to build

[1]This section describes an earlier method of learning from keyframe demonstrations. Later on in the thesis, Hidden Markov Models are used as described in Chapter 6

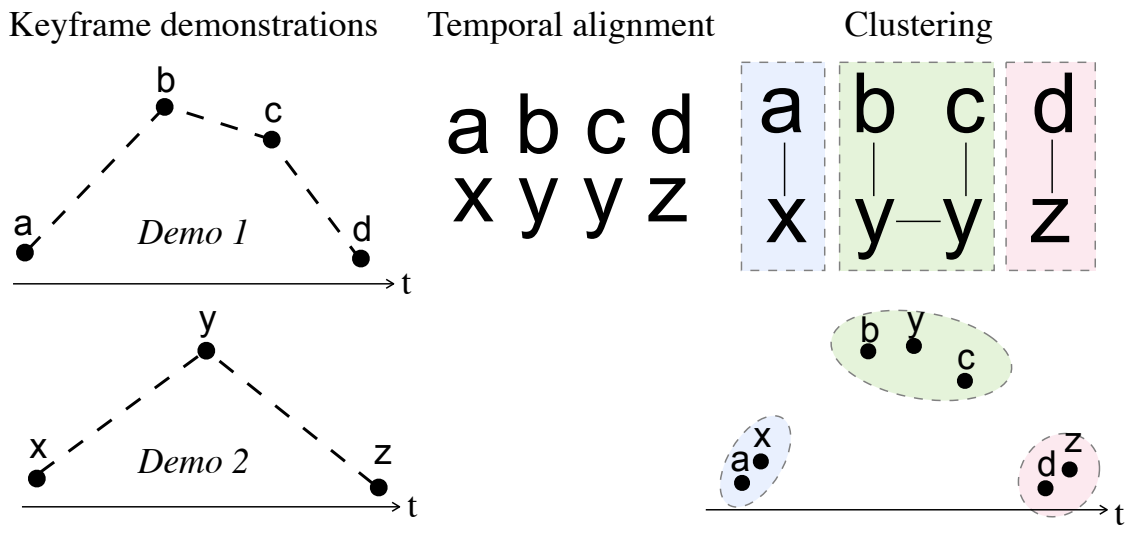


Figure 19: Illustration of the alignment and clustering process.

a model.[2] Dynamic Time Warping (DTW) is a widely used method for aligning two sequences at a time. In LfD, most of the time there are more than two demonstrations to be aligned. As a result, a general and order-independent method for aligning multiple keyframe demonstrations is needed.

The method developed in this section keeps an average alignment to which all the sequences are aligned in an iterative process and keep an alignment pool (a set of previously aligned sequences). The average and the alignment pool are initialized with the lowest cost pair. After that, the next sequence is selected based on the lowest pairwise DTW cost between the aligned and not aligned sequences. The average and the pool are then updated with this sequence. This process is repeated until all the sequences are aligned.

After aligning, the method clusters together any keyframes that are aligned to the same keyframe from another demonstration. This can be considered as finding connected components in a graph which connects all keyframes that are aligned together through DTW. An illustrative example is given in Fig. 19. The outcome of this step is the learned action model, which corresponds to the keyframe means and covariances of each cluster

[2]Not all LfD techniques have this problem, *e.g.* [40].

in sequence. Fig. 18(c) shows the clusters formed from the keyframe demonstrations in Fig. 18(b).

5.1.3 Skill Reproduction

Given the action model, fifth order splins are used to reproduce the learned skill. The spline is used to calculate states (*e.g.* positions) given time. This is motivated by the work in [31], which showed that human point-to-point motions resemble minimum-jerk trajectories and a fifth order spline is the function that minimizes the jerk cost and by the work in [70] which applies the same principles to motions with more than two points.

A spline is fit between two keyframe clusters. A fifth order spline has 6 unknowns. The positions, velocities and accelerations at the cluster means are used to calculate these unknowns. When there is no velocity information, *i.e.* clusters made of pure keyframe demonstrations, zero velocity and acceleration is used. For trajectory demonstrations, the mean velocities and accelerations at the cluster centers are calculated from the demonstrations. The other component is the duration between two keyframe clusters. The average duration seen in the input trajectories are used for this purpose. This splining results in C^2 continuity at the keyframes and C^∞ elsewhere[3].

In Fig. 18(d) we show the trajectory reproduced with the described method from the model in Fig. 18(c). Note that there seems to be non-smooth transition on some of the keyframes on the generated letter *P*. This is due to the low velocity and acceleration seen in the demonstrations.

5.2 Evaluation Domains

This section describes the two domains used for evaluating the KLfD method and the evaluation metrics

[3] C^k continuity for a function means that the function's $1 \ldots k$ derivatives exist and are all continuous

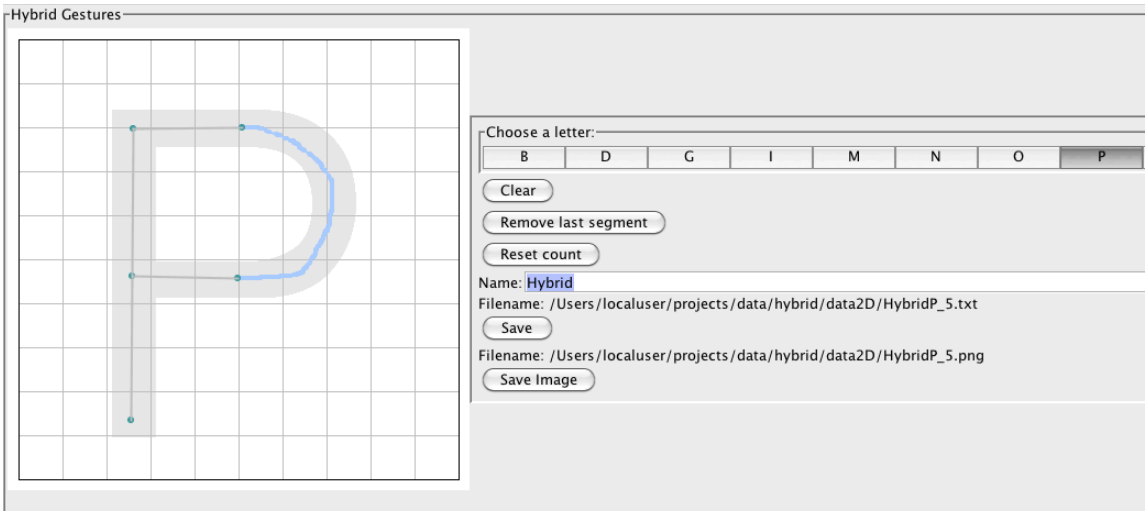


Figure 20: Snapshot of the Java applet for collecting 2D mouse gesture data. The target letter to demonstrate is shown as a light grey template that is 38 pixels thick.

5.2.1 Letters in 2D

Part of our evaluation is performed with 2D mouse gesture data, collected with a Java Applet (Fig. 20). The applet allows for collecting all three types of demonstrations; trajectory (TD), keyframe (KD) and hybrid (HD). A single click on the applet creates a keyframe, while dragging the mouse results in a trajectory segment.

5.2.1.1 Skills

This section evaluates six different skills corresponding to the letters: B, D, G, M, O, and P. The letters were chosen to have a variety of combinations of straight and curved segments. For each skill an image is created that consists of the template of the letter. The template is a light gray image of the letter with an average thickness of 38 pixels.

5.2.1.2 Success Metric

The goal of the skills in the 2D domain is to stay as close to the center, or skeleton, of the letter template. The ground truth skeleton is determined automatically as follows: The template is converted to a binary image and morphological *thinning* operation is applied to it. This creates a one pixel thick skeletal image (e.g., the red line in Fig. 24). Next, a

starting position is chosen on the skeleton that roughly matches where the demonstrator begins their demonstrations.

The success metric for generated trajectories is the DTW alignment cost between generated trajectory and the skeleton goal path, normalized by the length of the generated trajectory. A modified depth-first search algorithm is used to create a path given the skeletal image to create the ground truth. Pixels on the skeleton are added based on a depth-first search which explores neighboring skeleton pixels clockwise starting from the bottom-left one. Starting from the initial pixel, points are added to the trajectory when a pixel is first explored and when backtracking leads to a pixel. The search concludes when all the skeletal pixels have been explored.

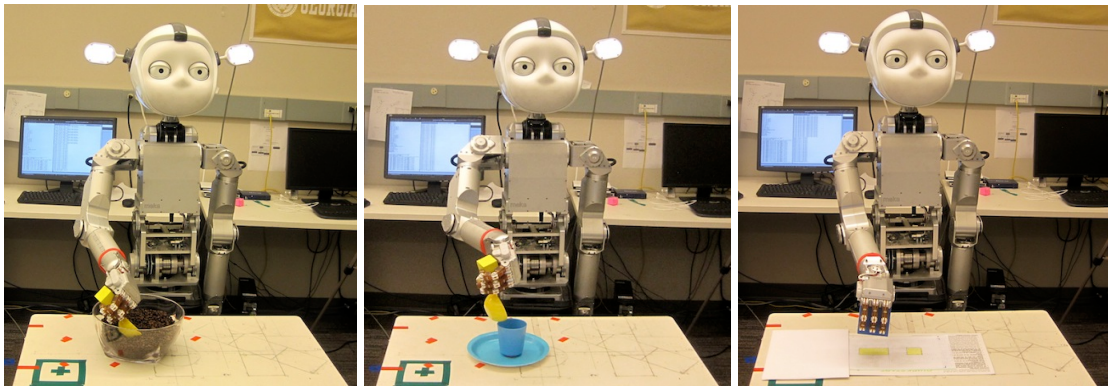
Since the generated trajectories might have variable velocity and the goal trajectory has constant velocity, the generated trajectory is re-sampled so that any two consecutive trajectory points are separated by the same distance which is set to be one pixel.

5.2.1.3 Data Collection

The data is collected through the applet shown in Fig. 20 on a MAC PC using a generic USB optical mouse. Four demonstrations were collected with each demonstration type (TD, KD, HD) for each letter. The hybrid demonstrations were chosen based on intuition: straight portions were shown as keyframes and curved portions were shown as trajectories (*e.g.* see Fig. 25). All demonstrations started at the same point for each letter, based on intuition on the starting position that would be optimal for drawing the letter in one continuous motion. This corresponds to the leftmost of the bottommost pixels, except in the case of G, which is drawn starting from the topmost endpoint. All demonstrations were provided by one of the authors in [3].

5.2.2 Robot Skills

In the second experimental domain, the approach is evaluated with table top manipulation skills on Simon.



(a) Scoop (b) Pour (c) Place

Figure 21: The three robot skills used in our evaluation.

5.2.2.1 Skills

This experiments uses he following three skills for evaluation:

- *Scooping:* In this skill, the robot holds an empty spoon and the teacher guides the arm to scoop as many coffee beans from a bowl as possible in one demonstration (Fig. 21(a)). This skill is demonstrated in trajectory mode. The success metric for scooping is the amount of coffee beans scooped (in grams).

- *Pouring:* In this skill, the robot holds a spoon full of coffee beans and the teacher guides the arm to pour as many beans from the spoon to a cup as possible in one demonstration (Fig. 21(b)). This skill is demonstrated in trajectory mode. The success metric for pouring is the amount of coffee beans successfully transferred into the cup (in grams). The initial content of the spoon is always the same.

- *Placement:* In this skill, the robot holds a block and the teacher guides the arm to place it to a designated area (Fig. 21(c)) with KD.

5.2.2.2 Data Collection

The setup for collecting demonstrations is illustrated in Fig. 22. Each skill is demonstrated for three goal locations (the bowl location for scooping, the cup location for pouring and

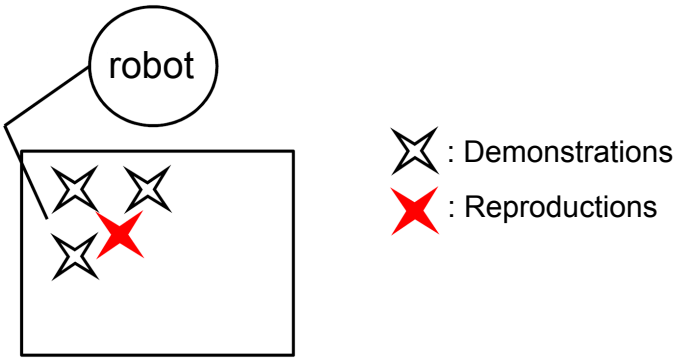


Figure 22: Setup for data collection and evaluation on the robot.

the designated target area for placement). Two demonstrations per location are recorded, resulting in a total of 6 demonstrations per skill. All demonstrations were provided by one of the authors. A different goal location is used for the evaluation of the reproduced skill.

The state recorded during demonstrations is the end-effector pose, represented by a 7D vector consisting of a 3D vector for positions and a 4D vector for rotations in the form of a unit quaternion, with respect to the target object. The trajectory segments are filtered using a Gaussian filter with the cut-off frequency chosen as $2Hz$. The filtering is necessary since the teacher demonstrations are inherently noisy. The frequency is chosen empirically based on the frequency amplitude spectrum of the data.

5.3 Evaluation

This section provides qualitative and quantitative evaluations of the KLfD learning framework. The evaluations start with example executions of the learned models on both domains, followed by comparison of KLfD with an LfD method on trajectory demonstrations. Finally, the learned models are compared when used with three different types of input demonstrations (TD, KD and HD).

As a baseline for comparison, the LfD method described in [21] is chosen. In this method, a Gaussian Mixture Model (GMM) is fit to the data using the Expectation-Maximization (EM) algorithm. Then, Gaussian Mixture Regression (GMR) is used for skill reproduction from the model. There are multiple reasons for this baseline choice. This method can be

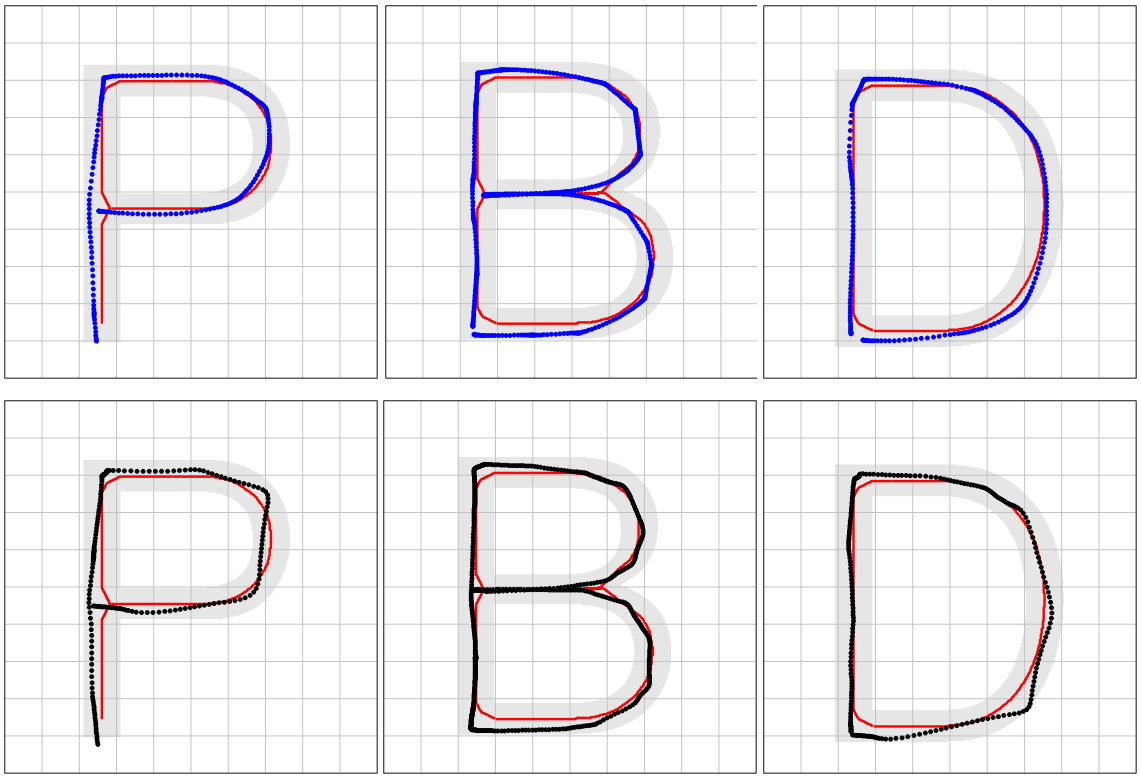


Figure 23: Letter reproductions using the KLfD (top row) and a baseline trajectory learning approach (GMM+GMR) (bottom row) with trajectory demonstration inputs for 3 skills in the 2D letter domain. The thin red line shows the skeleton of the letter that the teacher tries to demonstrate using the mouse. The thick lines show the reproduced trajectory.

trained with a low-number of demonstrations. The GMR portion generates smooth trajectories. This method is referred to as GMM+GMR. The trajectories are aligned in time using DTW prior to being input to this algorithm.

5.3.1 Sample Executions

5.3.1.1 Letters

Fig. 23 shows the reproduced skills that result from KLfD and GMM+GMR for trajectory type input demonstrations on a subset of three letters in the 2D domain. Note that there seems to be a piece-wise linear effect. This is due to low velocity and acceleration and sharp turns inherent in the provided demonstrations (see Fig. 18(a)). Both approaches produce qualitatively similar letters given the same trajectory input data.

Only KLfD is used to learn models from the keyframe and hybrid demonstrations since

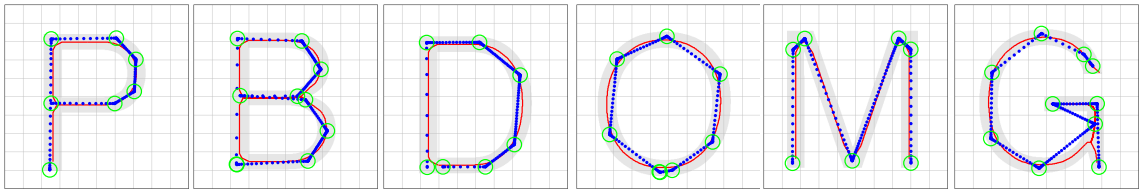


Figure 24: Letter reproductions using the KLfD with keyframe demonstration inputs for 6 skills in the 2D letter domain. The thin red line shows the skeleton of the letter that the teacher tries to demonstrate using the mouse. The thick lines show the reproduced trajectory.

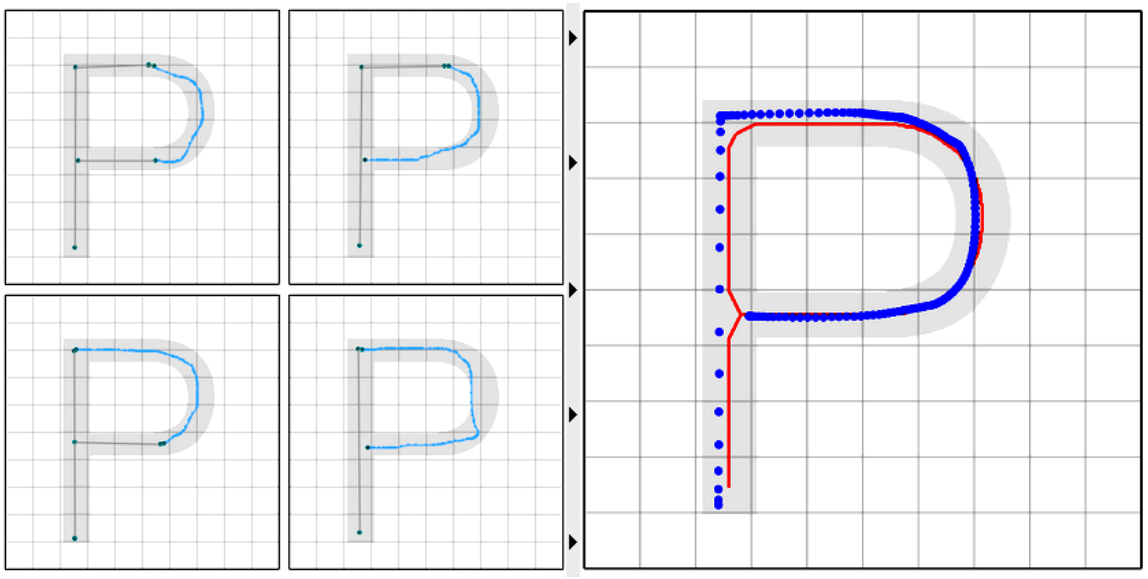


Figure 25: Reproduction the letter P with the KLfD for hybrid demonstration. The red line shows the skeleton of the letter and the blue dots show the trajectory reproduced based on the learned skill.

the GMM+GMR baseline approach is not designed to handle these. Fig. 24 shows the outcomes for keyframe demonstrations on all six letters in the 2D domain. Note that these models are obtained from multiple demonstrations (4). It can be seen that the resulting trajectories resemble the intended letters. The piece-wise linear appearance is due to our zero initial and final velocity assumption on keyframes. Comparing Fig. 24 and the top row of Fig. 23, it can be seen that the trajectory demonstrations result in learned models that look more similar to the intended letters for certain letters, since the demonstrations themselves contain more information about the curved parts.

Fig. 25 shows the set of four hybrid demonstrations provided for the letter P and the resulting reproduction. The KLfD method succeeds in learning an appropriate model, despite the non-uniformity of the demonstrations. It can be argued that the resulting letter P is more similar to the intended one than any of the Ps in Fig. 23 or Fig. 24.

5.3.1.2 Robot skills

Fig. 26 shows the demonstrations provided for the scooping skill and the trajectories reproduced using GMM+GMR and KLfD models. Two representative dimensions of the state-space are shown: the vertical dimension and the angle-component of the quaternion. The top row corresponds to pre-processed teacher demonstrations and the extracted keyframes. Note that the data is highly varied and not aligned in time. The middle row shows the aligned trajectories (as described in section 5.1.2), the learned GMM and the resulting trajectory. The bottom row shows the aligned keyframes, the KLfD model and the resulting trajectory. The algorithm for alignment is the same for trajectories and keyframes but the input data is different.

The vertical dimension (left column in Fig. 26) of the scoop captures the dip into the bowl. It can be argued that the variance is lower in the dipping portion. This is from the fact that all the demonstrations had this in common, *i.e.* this was the important part of the skill. Note that both of the methods generated similar resulting trajectories and captured

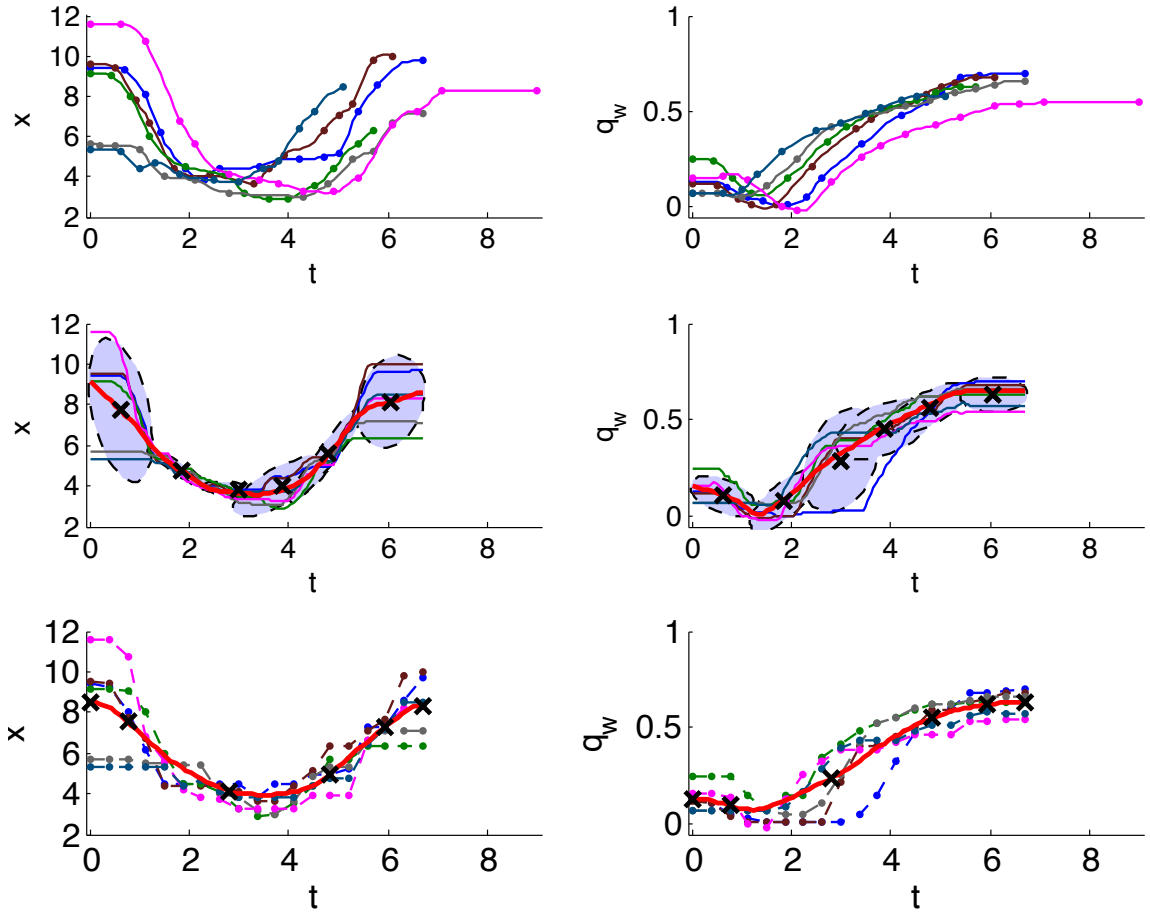


Figure 26: The demonstrations and the learned trajectories for the x (verticals)) and the q_w (angle representation of the quaternion) dimensions of the scoop skill. Vertical axes correspond to the dimensions and horizontal axes correspond to time. Top row: Filtered and transformed (with respect to the object) raw trajectories and the extracted keyframes (dots). Middle Row: Aligned demonstrations and the learned trajectory (red) using GMM+GMR. The covariance between the dimensions and time is represented by the light blue ellipsoids and x-marks represent the centers of the GMMs. Bottom Row: Aligned keyframes (dots, dashed lines are to ease visualization) and the learned trajectory (red) using the KLfD method. The x-marks denote the means of the pose distributions.

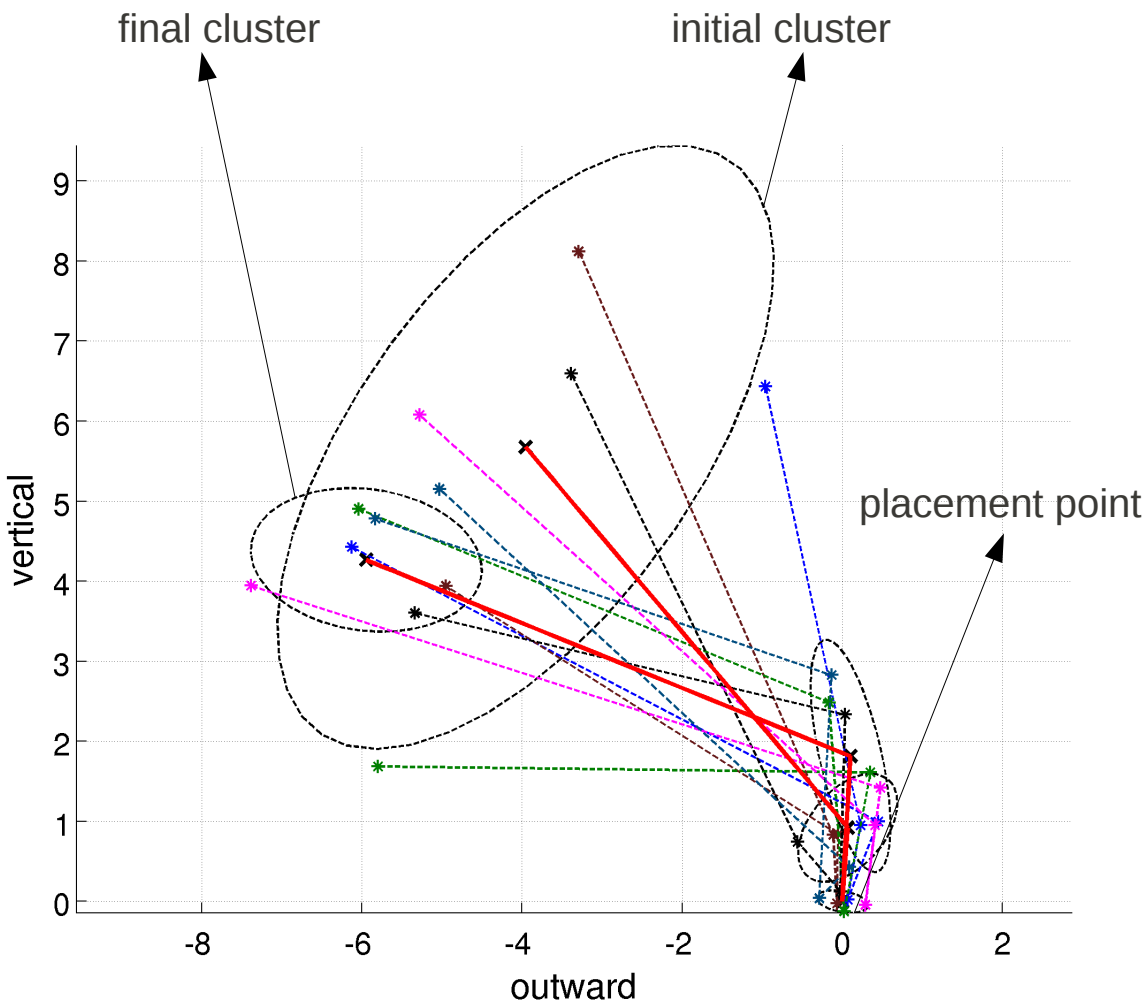


Figure 27: The 2D projection of the placement demonstrations. The asterisks mark the demonstrated keyframes. The dashed-lines are given for visualization purposes. The ellipses represent the covariances and x marks represent the means of the pose distributions and the red solid line is the reproduced trajectory.

the low variance of this portion.

A rotation quaternion represents a rotation around an axis. Specifically, the angle-component is the cosine of the half of the rotated angle. A single component of the quaternion by itself is not enough to capture all the rotation information of the end effector but gives a rough intuition. The resulting trajectories (right column in Fig. 26) show that there is nearly a monotonic change in this angle which is consistent with the scooping skill.

Fig. 27 shows the 2D projection of the keyframe demonstrations, the resulting skill model learned with KLfD and the generated trajectory for the placement skill. Note that

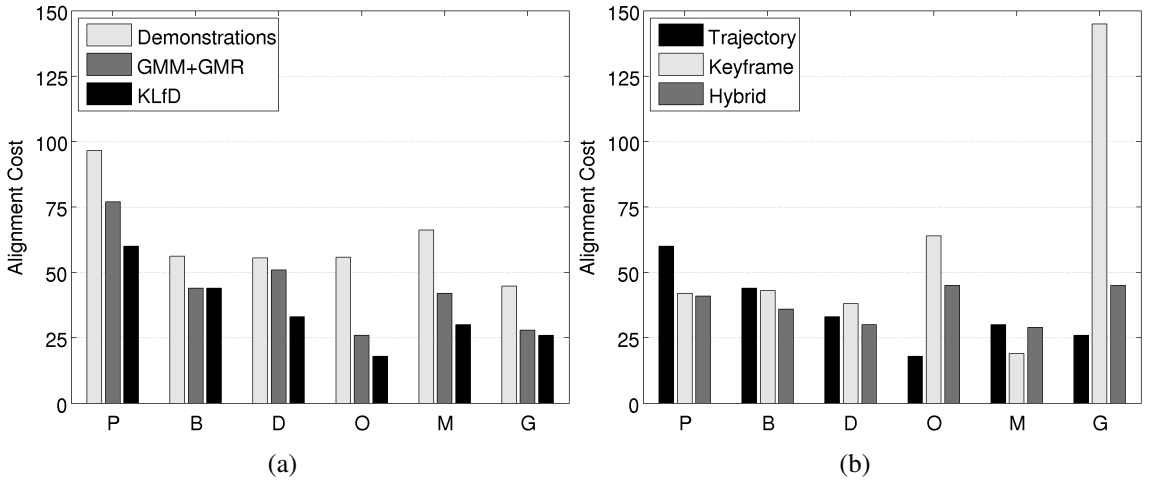


Figure 28: Skill success in the 2D letter domain measured with costs for alignment with the template letter (lower cost means better alignment). (a) For skills learned with GMM+GMR versus with KLfD using trajectory type input demonstrations. (b) For skills learned with KLfD using three different input demonstration types. Note the KLfD bars in (a) are equivalent to Trajectory bars in (b).

the initial and final clusters have higher variance and variance lowers as the skill approaches the placement position. The algorithm identified 5 important regions for the skill. These are interpreted as the start and end of the skill, pre-placement position, safe retraction position and the placement position. During the placement demonstrations, the teacher was able to take his time to correctly align the block with the placement position, which was possible to due to the keyframe demonstrations.

5.3.2 Comparison with trajectory-based methods

The framework accommodates trajectory demonstrations by converting them to keyframe demonstrations, as described in section 5.1.1. This can be viewed as a loss of information. In order to show that this loss does not effect the performance of the learned skill, the initial focus is on trajectory demonstrations as the input. This is done in order to quantitatively evaluate the KLfD framework in comparison to the baseline GMM+GMR.

Table 6: Comparison of the success of (i) provided demonstrations, (ii) trajectories learned with KLfD and (iii) with GMM+GMR on two skills. Values indicate weights in grams and standard deviations are given in parentheses. Note that demonstrations have 18 samples and learned models have 10.

	Scoop	Pour
Demonstrations	38.4 (7.3)	26.2 (5.8)
Learned skills (KLfD)	41.5 (2.0)	23.0 (1.7)
Learned skills (GMM+GMR)	37.8 (1.4)	27.8 (2.3)

5.3.2.1 Letters

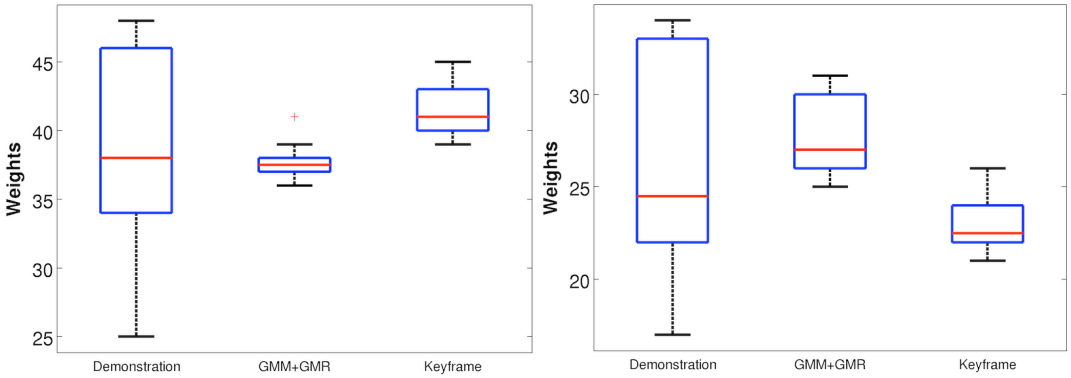
A comparison of performance on the six letter skills is shown in Fig. 28(a). The success metric is alignment cost, as described in Sec. 5.2.1. For all six letters the models learned with KLfD produce letters that are closer to the template (*i.e.* have lower alignment cost with the template skeleton). In addition, both methods produce skills that are more successful than the provided demonstrations of the skill.

5.3.2.2 Robot Skills

KLfD and GMM+GMR are compared on two robot skills: scooping and pouring. The success metric is the bean weight, as described in Sec. 5.2.2. Demonstrated trajectories were played back on the robot three times per respective location (a total of 18 demonstrations) and the success metric is recorded for each. These results can be seen in Table 6 in the *Demonstrations* row. This is a sanity check on the data, showing that robot has seen successful demonstrations, so expectation is that the learned models to perform similarly.

The reproduced skill is performed at a different target location (the red cross in fig. Fig. 22). Each learned model is executed 10 times. The results are reported in Table 6 and the descriptive statistics of the results can be seen as box-plots in Fig. 29.

These results show that the performance of both learning methods have success similar to the demonstrations. Moreover, they are similar to each other. For the scooping skill KLfD resulted in a more successful learned model whereas GMM+GMR did better for the pouring skill. The methods were not tuned with respect to any skill and parameters were



(a) Weight of scooped coffee beans (b) Weight of poured coffee beans

Figure 29: Box-plots for the skill success measures comparing (i) replayed teacher demonstrations (18 samples), (ii) trajectory obtained with the model learned with the GMM+GMR method (10 samples) and (iii) with the KLfD method (10 samples) for two skills.

chosen to be generic. This shows that the KLfD method is on par with a standard technique at building models from trajectory demonstrations.

5.3.3 Comparison of demonstration types

Next, the impact of input type (TD, KD, HD) on skill success in the 2D letters domain is evaluated. This comparison, in terms of the alignment costs with the template skeleton, is shown in Fig. 28(b). Hybrid demonstrations result in the best performance for the letters P, B and D, followed by keyframe demonstrations. Hybrid demonstrations have an advantage over keyframe demonstrations due to the ability of using trajectories for the curved parts of the letters. Trajectory demonstrations have the highest costs in these skills. This is mainly due to the difficulty of drawing a straight line when giving trajectory demonstrations.

For the letter O it can be seen that trajectory demonstrations result in the best performance. This is again intuitive since this letter is entirely curved. At the other end of the spectrum, the drawing of letter M consists of only straight movements. As a result, we find that a pure keyframe demonstration results in the best alignment. For the hybrid demonstrations of these two letters, we intentionally tried to balance the use of keyframe and trajectory segments, even though the usage of hybrid demonstrations for these letters

is less intuitive. For the letter G we see that trajectory demonstrations perform best, since the letter is predominantly curved.

Overall we see that the best KLfD performance results are achieved when the demonstration type is suited for the skill. In the 2D letter domains this implies that using trajectory demonstrations for O and G, keyframe demonstrations for M and hybrid demonstrations for P, B and D. This confirms our intuition about the utility of being able to provide a variety of demonstration types to handle a range of skills.

5.4 Summary

This chapter developed a framework to learn from hybrid demonstrations called *Keyframe-based Learning from Demonstration* or KLfD, based on the results of the experiment described in Chapter 4. This framework can handle trajectory, keyframe and hybrid demonstrations in a unified manner. The KLfD idea is based on converting all types of demonstrations into keyframes.

This allows a human teacher to use the input mode that is most comfortable to them or that they see most suitable for a given skill. In addition, this allows them to change their input mode over time, *e.g.* show some trajectory demonstrations and some keyframe demonstrations for the same skill.

Hybrid demonstrations are particularly strong as they allow the demonstration to be adapted to the particular parts of a skill. Typically skills involve multiple components. For instance it is natural for scooping and pouring to be demonstrated together. Parts of the skill that requires a complex motion of the spoon to collect the beans or to pour them accurately into the cup are suited for trajectory demonstrations. Whereas, the parts before, after or in between these movements are more suited for keyframes. This is analogous to the 2D skills corresponding to the letters P, B, D we considered in Sec. 5.3.3. KLfD produces the best results with hybrid demonstration inputs for these skills. The hybrid demonstrations allow for traditional trajectory demonstrations, so there is an added benefit

with hybrid demonstrations instead of a trade off. The results showed KLfD can learn skill models that have performance on par with existing methods. This implies that KLfD is a viable alternative even for conventional demonstration types, while accommodating new demonstrations types.

The results from the experiment described in Sec. 4.3 for teleoperation and this chapter, and the anecdotal observations from the AAAI 2011 LfD challenge [5] suggest that hybrid demonstrations are a valuable addition to learning from demonstration.

The specific learning method, described in Sec. 5.1.2 which corresponds to the *Temporal Alignment and Clustering* box in Fig. 17 has some drawbacks. It does not utilize the covariance of the clusters. It cannot learn a branching structure and as a result end-up averaging all the demonstrations which may not be desirable. There is also the risk of ending up with less number of clusters than required if the number of provided keyframes vary too much between demonstrations. This part of the KLfD framework can be replaced by a Hidden Markov Model approach described in Chapter 6.

This chapter developed a method to learn from hybrid demonstrations. However, it did not take advantage of the fact that teachers are goal-oriented, as observed in Chapter 4. Starting from the next chapter, this thesis will concentrate on leveraging this goal-oriented behavior of the non-expert teachers in the context of LfD.

CHAPTER VI

ACTION AND GOAL MODELS

The experiments described in Chapter 4 under various conditions had one thing in common: teachers tend to concentrate on achieving the goal of the demonstrated skill rather than on consistent demonstrations of how to achieve it. The teachers paid more attention to demonstrating successful instances of the skills. Majority of the teachers were fine with noisy, inconsistent and unnecessary arm motions as long as the skill was demonstrated successfully, whereas only a handful paid attention to **both** aspects. This suggests that naïve teachers try to communicate the goal of the skills more so than the exact actions to achieve that goal.

This observation is a pivotal point in this thesis. The aim of Experiment I and II was understanding how non-expert provide demonstrations to the robot. At that point, demonstrations were only interpreted as actions. Some users had severe trouble demonstrating these which was frustrating since users were happy as long as they showed a successful instance of the skill at the end. This was very difficult from an algorithmic point of view since the data was not suitable for learning. Hybrid demonstrations and the KLfD method were attempts to remedy this. However, the observation about the goal-oriented nature of people during teaching changed the way this thesis looked at non-expert demonstrations completely. In retrospect, human teachers being goal-oriented is quite logical and agrees with a vast literature in developmental psychology pointing to the fact that humans are goal-oriented in their perception and imitation of motor skills from a very early age [25, 50]. From this chapter and on, this will be the main focus.

Motivated by the goal-oriented nature of people, this chapter introduces simultaneous

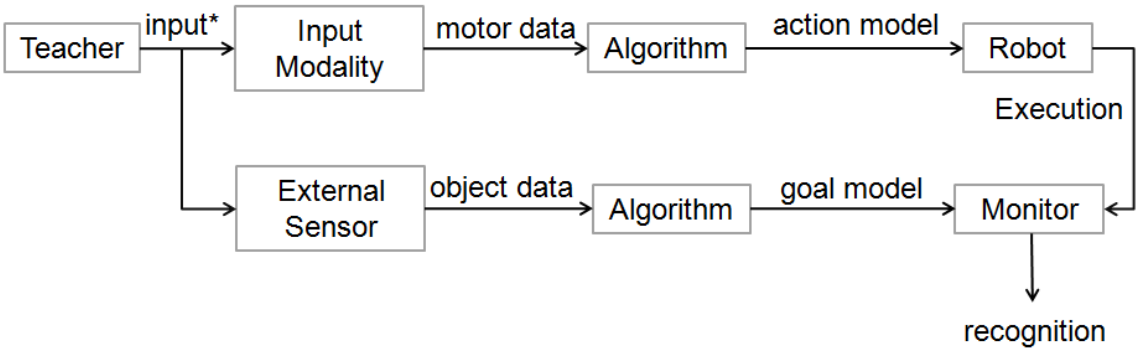


Figure 30: The version of the LfD system to learn action and goal models that shows the GoaLfD framework. The user demonstrates the skill by using keyframes. Two types of data is extracted at each keyframe; motion data (related to robot control) and object data (related to the object being manipulated for the skill). Then the same algorithm is used to learn two distinct models from the aforementioned data; an action model and a goal model. The learned action model is used to execute the skill and the learned goal model is used to monitor the execution.

learning of an action model and a goal model for a skill from the same set of demonstrations, while maintaining them as separate learning problems. This frame work is called the *GOal and Action Learning from Demonstration - GoaLfD*. This is a novel approach to skill learning in LfD where typically the problem is defined as learning in a motion space (*e.g.* joint space, end-effector space) or in a combined sensorimotor space that is known in advance to be good for representing a particular skill. An action model captures the how-to-do part and a goal model captures the what-to-do part of a skill. The learned action models are used to execute the skills and the goal models are used to monitor these executions. Note that action and goal models are complementary and are inherently tied together by the robot and its environment.

The approach developed in this chapter uses keyframes as demonstration inputs. The keyframe demonstrations complement goal learning very well by allowing the teachers to highlight salient parts (*e.g.* sub-goals) of the skills while ignoring the unnecessary bits. The work described in this chapter is published in [9].

6.1 Overview of the Learning Approach

The approach presented in this chapter is illustrated in Fig. 30. Keyframe demonstrations are used as input from the user. Keyframes help the users highlight salient points of skills which is very helpful in learning goals. The drawback is that keyframes lose any dynamics information. Since the demonstration pauses at each keyframe, the robot does not receive any information about target velocities and/or accelerations. However, the presented action and goal learning approach concentrates on object-based manipulation skills in which dynamics is not a component of the goal. The purpose of these skills is to achieve certain object states during the skill. This class of skills encapsulates many day-to-day activities (e.g., fetching items, simple kitchen tasks, general cleanup, aspects of doing laundry or ironing *etc.*). The approach and its current implementation imposes several other assumptions on the types of skills that can be handled. There assumptions are that skills have a single object of attention, they can be executed by a single end-effector, they are not cyclic and their goal involves a perceptual change in the object.

The main idea of the approach is to use two different information streams, from the same set of demonstrations, to learn two different models, as seen in Fig. 30. A single demonstration involves the teacher marking a series of keyframes that complete the skill. Each time the teacher marks a keyframe, two types are recorded: (1) an *action keyframe* consisting of motion data, and (2) a *goal keyframe* consisting of object data. Thus a single demonstration is treated as two simultaneous demonstrations: the *action demonstration* is the set of action keyframes and the *goal demonstration* is the set of goal keyframes. The approach uses the same learning algorithm on each type of data to learn the two different models.

There are multiple reasons for separating action and goal learning. Data for both are inherently from different sources; the action data comes from the robot and the goal data comes from the object of interest. This separation allows for different levels of granularity between the models. The action might require multiple steps to change to state of the

object. For example to lift a cup, the robot would need to approach the cup grasp it and then lift it. From the goal point of view, the object only changed its height. In addition, the learned models have different purposes. The action model is used to execute the skill. The goal model is used to monitor the execution of the skill. The monitoring task involves using the goal model to classify the object data stream captured during skill execution.

The purpose of the GoaLfD is to handle a variety of skills without tuning and to represent the acceptable variance on executing the skill as opposed to a single optimal way. Relatively generic feature spaces are needed to handle a variety of skills. This requires the approach to learn from multiple demonstrations so that there is enough information about the variance over how to execute the skill. Having a variance on the action model allows the robot to execute the skill with variety which is important in cluttered and/or new environments. Simply repeating a single demonstrations might fail or might not be possible. An example to the latter is when a transformed demonstration falls out of the robot's workspace due to a new object location. The allowable variance over the execution is also useful for avoiding collisions in clutter but this is not addressed in the current chapter[1]. In addition, having multiple demonstrations allows to estimate the variance on how the skill *looks*, important for monitoring, especially given that sensors are noisy. A single demonstration is not enough to build a good goal model; even the same exact repetitions will not look the same from the sensor's perspective.

In the remainder of this chapter, the implementation of the system depicted in Fig. 30 will be described.

6.2 State Spaces of the Models

6.2.1 Object Data

In selecting a feature space for the object data, the aim is to have a feature space that is going to allow the robot to build a visual model of how the object changes over the

[1]The work described in [46] is a preliminary step in this direction.

course of the action. Thus, this approach selects a set of features commonly utilized in the literature for object tracking and perception tasks; a combination of location, color, shape and surface features. As advances in object perception are made, this object feature space can be updated to reflect the state of the art.

An overhead RGBD camera is used as the external sensor. As a result, the raw sensor information for the object data is the colored point cloud data. The goal keyframe consists of features extracted from this RGBD data. Two assumptions are made about the objects and the environment: (1) the objects sit on a plane (*e.g.* tabletop) and (2) the objects have relatively solid color. The objects are segmented using the approach in [71] to find spatial clusters of similar color. This procedure often results in over segmentation, especially if the object is occluded. Another pass is taken at the cluster level to merge the similar ones.

As stated in Sec. 6.1, this approach assumes that there is a single object of attention for each manipulation action to be learned. In the implementation presented here, selecting this object is simplified by using a clean workspace in which the object of attention is clearly visible and its color known a priori. In future work, this could be selected automatically based on which objects the hand moves closest to, or which objects changed the most over the action, or by interacting with the teacher *etc.*

After segmentation, a rotated bounding box is fit to the object. The pose of this box is used as the object pose. An example of the segmentation and the bounding box results can be seen in Fig. 33. Then the method extracts color, generic shape and surface related features from the object using the point cloud and bounding box data. Some skills can change the object location (e.g. pick and place) and color (e.g. pouring a different colored liquid in to a cup). Remaining features are extensions of commonly used 2D features. They represent the generic silhouette of the object, which can be changed by certain skills. Overall these are more global features for the object. The View Point Feature Histogram (VFH) descriptors [63] are used as the surface features. These features have been shown to work for object recognition. The first step to calculate these features is to estimate the

surface normals at each point followed by finding the centroid of the points. Then, the angular deviations between the axes of the surface normals at each point to the centroid of the object is calculated and binned to form a histogram. The reference implementation in PCL is used to calculate the VFH features. In this thesis, different versions of this feature space is used. The exact features used will be detailed at the appropriate parts.

6.2.2 Motion Data

The end-effector with respect to the target object is used as the motion data, which constitutes the action keyframe. After a demonstration, end-effector poses are transformed to the object reference frame (as calculated in Sec. 6.2.1). This object based representation is fairly common in robotics. The end effector pose is represented as the concatenation of a 3D vector as the translational component and a unit quaternion (4D) as the rotational component, resulting in a 7D vector. Hence the action model lives in a 7 dimensional action space, treated as $\mathbb{R}^7$. A point in this action space is projected onto the space of rigid body transformations, SE(3), by normalizing the quaternion part wherever necessary (*e.g.* before execution).

6.2.3 Notation

In this section a list of symbols is defined for several of the constructs introduced so far, to be used in the rest of the text.

a_i^j**:** The i^{th} action keyframe for the j^{th} demonstration, $a_i^j \in \mathbb{R}^7$

g_i^j**:** The i^{th} goal keyframe for the j^{th} demonstration, $g_i^j \in \mathbb{R}^{43}$

A^j**:** The j^{th} action demonstration, a set of action keyframes where $m(j)$ is the number of keyframes (the number of keyframes can be different for each demonstration), $A^j = \left\{a_1^j, a_2^j, \ldots, a_{m(j)}^j\right\}$.

G^j**:** The j^{th} goal demonstration, a set of goal keyframes,
$G^j = \left\{g_1^j, g_2^j, \ldots, g_{m(j)}^j\right\}$.

D_A**:** The set of n action demonstrations, $\left\{A^1, \ldots A^n\right\}$

D_G**:** The set of k goal demonstrations, $\left\{G^1, \ldots G^k\right\}$. k and n can be different

q_r**:** The r^{th} observed keyframe during a skill execution, $q_r \in \mathbb{R}^{43}$.

Q**:** The set of observed keyframes during a skill execution, $\left\{q^1, \ldots q^p\right\}$, where p is the number of keyframes used in execution.

M_A**:** The action model

M_G**:** The goal model

6.3 Learning the Models

This approach uses Hidden Markov Models (HMM) to represent both the action model and the goal model of the skills. HMMs are useful tools for modeling sequential data where observations are noisy and sample independence assumption is too constrained. Keyframe demonstrations lend themselves naturally to such a model since they can be treated as sequential observations that are not independent. In addition, HMMs are generative which enables the approach to use them in skill execution.

The emissions are modelled as multivariate Gaussian distributions on the corresponding state space (either the action space or the goal space). The HMM notation used consists of the following:

N **:** The number of states

s_j **:** The j^{th} state ($j = 1 \ldots N$). The states are not directly observable.

S **:** The set of all states, $S = \{s_1, \ldots, s_N\}$

μ_j**:** The emission mean for the j^{th} state

Σ_j**:** The covariance matrix for the j^{th} state

T**:** The $N \times N$ state transition matrix,

$T(k, j) = P\left(s(t) = s_k | s(t-1) = s_j\right)$ is the transition probability from state j to state k

y**:** An emission vector

$P(y|s_j)$**:** The probability for the emission y in state s_j,

$P(y|s_j) \sim \mathcal{N}(\mu_j, \Sigma_j)$

π**:** The N dimensional prior probability vector

ζ**:** The N dimensional terminal probability vector

The set of action demonstrations D_A is used to learn the action model, M_A. Similarly the set of goal demonstrations, D_G is used to learn the goal model, M_G. These models are learned from multiple demonstrations and both M_A and M_G are individual HMMs.

These HMMs are trained with the Baum-Welch algorithm (BWA) [13], which is an Expectation - Maximization (EM) type algorithm, initialized with k-means clustering, which itself is initialized uniformly randomly in the state space and restarted 10 times. The Bayesian Information Criterion (BIC) is used for model selection, *i.e.* to select the number of states of the HMMs. The model selection starts by setting the number of states as the minimum number of keyframes seen during the demonstrations and increases this number until minimum BIC score is hit. Then, the corresponding HMM is chosen as the model. Since the learning is initialized randomly, the approach runs BWA 10 times given a number of states and select the model with the highest likelihood to calculate BIC. Note that the action model and the goal model can have different number of states after training.

The standard BWA calculates T, π, $\mu_{1\ldots N}, \Sigma_{1\ldots N}$. In addition, the approach presented here calculates the terminal probabilities. The terminal probability of a state represents the

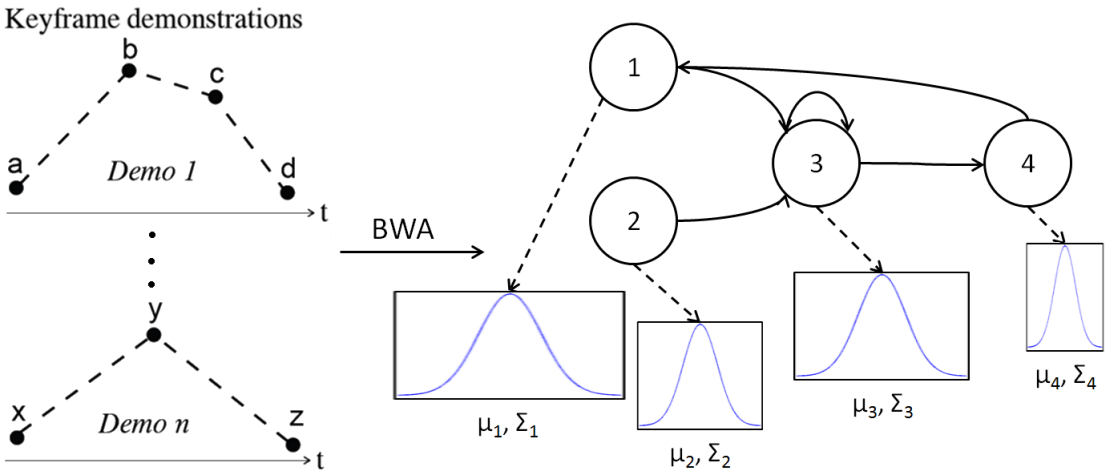


Figure 31: A depiction of the learning process. The resulting model is a representative 4-state HMM with emission distributions. The solid lines represent non-zero transition probabilities between states. In addition prior and terminal probabilities are learned.

likelihood of that state being the last state for the HMM, and is calculated analogous to the prior probability. The terminal probabilities are denoted with ζ. A representative learning process and the resulting HMM is shown in Fig. 31. Superscript is used to denote model membership for the parameters, for example π^A is the prior probabilities for the action model, T^G is the transition matrix for the goal model *etc.*

Another advantage of HMMs is that a single EM-step of the system, with the Gaussian emission models, is polynomial, *i.e.* tractable. This tractability is due to the Markov assumption of state transitions, emissions depending only on the current state, and the tractability of Gaussian model parameter estimation.

6.4 Utilizing the Models

6.4.1 Action Execution

The learned skill is executed by generating a trajectory from the action HMM, M_A. The first step is to generate a state path by finding the maximum likelihood path between the prior and terminal states by using the transition matrix (T_A). The next step is to take the emission means along the state path, which are the end-effector poses with respect to the object, and transform them to the robot's frame. Then a 5^{th} order spline is between the transformed

poses to get a trajectory for the robot to follow. This process is detailed in Alg. 1.

The generation of the trajectory starts by finding the most likely path between each prior state in π^A to each terminal state in ζ^A and storing them (lines 1-11). The function *FindStatePath*(p, z, T^A) does the path finding between the state p and the state z, given T^A. It applies Dijkstra's algorithm on the negative logarithm of the entries of T^A as the edge weights. The shortest path calculated by using the addition of the negative log-likelihoods is equivalent to the one that would be obtained by maximizing the multiplication of the probabilities. Then the most likely path among these paths is selected, given the transition probabilities T^A, prior probabilities π^A and terminal probabilities ζ^A (line 12). If the initial position of the robot is important, only the paths from the prior state that is closest to this initial position can be used.

Algorithm 1 Tra = GenerateTrajectory(M_A)

1: $\Phi = \emptyset$
2: **for all** $p \in S^A$ **do**
3: **if** $\pi^A(p) \neq 0$ **then**
4: **for all** $z \in S^A$ **do**
5: **if** $\zeta^A(z) \neq 0$ **then**
6: $\phi =$ FindStatePath(p, z, T^A)
7: $\Phi = \Phi \cup \phi$
8: **end if**
9: **end for**
10: **end if**
11: **end for**
12: $\rho = \arg\max_{\phi \in \Phi} (loglik(\phi))$
13: $R \leftarrow \emptyset$
14: **for all** $s \in \rho$ **do**
15: $R \leftarrow \mu_s$
16: **end for**
17: Tra = Spline(R,v_{avg}, Δt)
18: **return** Tra

It might be the case that there is a cycle in the resulting path, for example when the user starts and ends the demonstrations at close enough robot poses. For the purposes of this thesis, it is assumed that the skills are not cyclic thus when a cycle is detected in the

generated path, the process stops. In this case, only a single cycle of execution is allowed. Another way to resolve this is to interact with the user and ask if the skill has a cyclic component and how many times (or until what condition) the cycle should be executed.

Once a state path is selected, the resulting emission means of the generated path in the same sequence is selected (lines 13-16) and transformed to the robot coordinate frame (skipped in the algorithm for clarity). Then the method fits a quintic spline between each of them (line 17). The robot follows the resulting trajectory in the end effector space to execute the skill. The transformation is done based on the current object pose, as estimated by the perception system. A constant average speed (v_{avg}) between two poses is used to decide on the timing of the spline. In addition, the initial and final velocities and accelerations are taken as zero between the poses. This results in a straight path in the end-effector space between two points. A given time step (Δt) dictates on the density of the trajectory in time.

6.4.2 Goal Monitoring

The goal model, M_G, is used to monitor the execution of the action model, M_A. This information could then be used by the robot for error recovery or to ask the human teacher for help in case of a failure or to move onto its next task in case of success.

To perform monitoring during execution, the robot extracts an observation frame (q) from object data in the goal space at each action keyframe it passes through. The action keyframes are at the emission means obtained as described in Alg. 1. This results in an observation sequence, $Q = \{q_1, \ldots, q_p\}$, where p is the length of the state path ρ that is calculated in Alg. 1.

A sequence of a short length can have a high likelihood score but it might not be enough to complete the skill. For example, observing an incomplete execution of a skill would yield a high likelihood but in reality it should be a failure since the skill is not completed. This is the reason that the terminal probabilities are estimated from demonstration data. On the other hand, it is not enough to just check whether the end-state is a terminal state for all

the skills. Sub-goals of the skill might be important to achieve it but not be visible at the end state. The forward algorithm is used to calculate the likelihood of the the observed sequence with the inclusion of the terminal probabilities, $p_s = P(Q;M_G)$, given the goal model. Then the skill is deemed successful if $p_s > \tau_s$ holds true where τ_s is a selected threshold.

In the current implementation, the monitoring decision is made at the end of execution. However, there is no technical limitation for it to be done as the skill is being executed. The likelihood of the current observation sequence can be calculated online and evaluated with a threshold. The only difference would be that the terminal state check would be done at the end of the execution. This could be used to determine early failure and show when the action failed.

6.5 Summary

This chapter developed a novel framework for LfD, learning task level goals and motor level actions simultaneously, but maintaining them as separate learning problems. Explicitly separating the learning problem into these two spaces leverages the fact that human demonstrators are going to be goal-directed, and good at showing *what to do*, while only a subset of those teachers may also focus on showing the robot good demonstrations of *how to do it*. The developed method uses object relative end-effector poses to learn action models and a generic perceptual feature space to learn the goal model. The framework is outlined in Fig. 30. The main contributions of this chapter are the insight of learning actions and goals simultaneously and the developed system along with the individual algorithms for its subcomponents.

The action and goal learning method is evaluated with non-expert users in Chapter 7. The results show that goal models that can monitor the skill executions with high success rate can be learned even if the learned action models are not as successful.

CHAPTER VII

EXPERIMENT III: ACTION AND GOAL LEARNING WITH PEOPLE

The main theme of this thesis is to let everyday people teach skills to robots. Chapter 6 introduced action and goal models along with the approach to learn and use them - GOALfD. This chapter presents the evaluation of GOALfD with naïve teachers. The results described in this chapter is published in [9].

7.1 Experiment Details

The robot Simon is used for this experiment. An overhead ASUS Xtion Pro LIVE (RGBD camera) is used as the external sensor with a view of the tabletop workspace seen in Fig. 32. The experiment presented here follows the protocol described in Sec. 3.2.

7.1.1 Feature Space for Goal Models

The general version of the feature space was introduced in Sec. 6.2.1. This section describes the feature space used in this experiment.

The features include average RGB (3), number of points in the cloud, centroid of the bounding box (3), rotation of the bounding box with respect to the table normal, bounding box volume, bounding box area, bounding box side lengths (3), aspect ratio parallel to the table plane, bounding box area to volume ratio (scaled down) and bounding box volume to number of cloud points ratio, resulting in 16 features. Regarding the VFH features, 9 bins per angle is used. In the end, the goal space ends up wi 43 $(16 + 9 \times 3)$ dimensional goal space, treated as $\mathbb{R}^{43}$, for the goal keyframe.

This experiment collected batch data. An expert went through all the point clouds to make sure that the perception system worked. If the perception system picked out the

Table 7: Additional speech commands to switch between demonstration modes

Command	Function
Watch me do it	Transition to goal-only demonstrations
I will guide you	Transition to kinesthetic demonstrations

wrong object (*e.g.* a piece of the end effector) or was not able to merge clusters in the event of occlusion, the expert intervened and fixed it.

7.1.2 Demonstrations

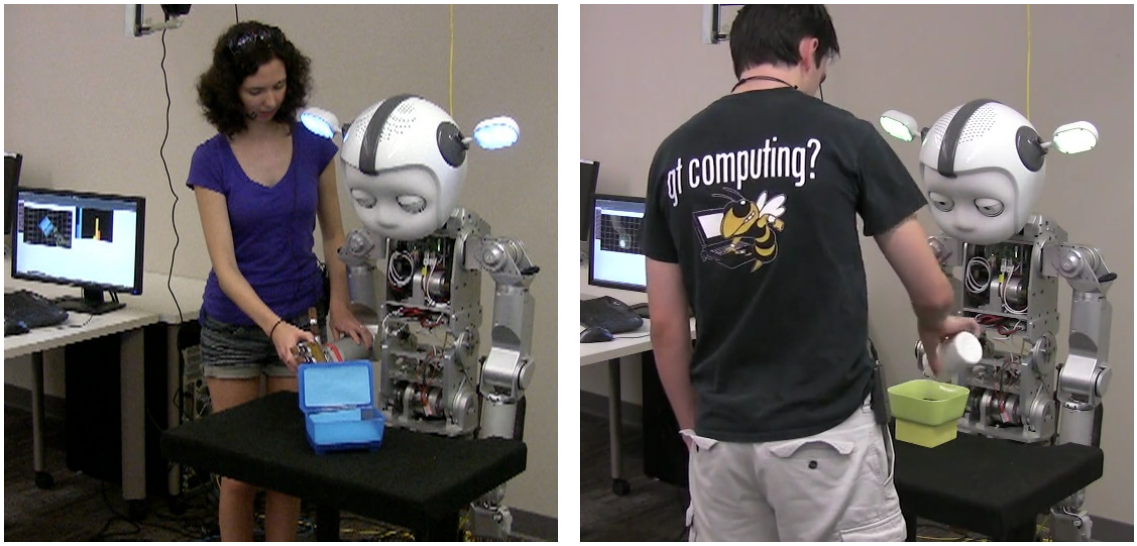
The teachers provide two types of demonstrations in this experiment. The first type is kinesthetic teaching, depicted in Fig. 32(a) as used in previous experiments. The second one is called *goal-only demonstrations*. The teachers provide demonstrations by doing the skill themselves with keyframes, as seen in Fig. 32(b). While this loses the action keyframes entirely, it does have the advantage that teachers can give demonstrations quickly and likely with less occlusions (due to only the human being in the view instead of human and the robot). In this case, when the teacher marks a keyframe, the robot only records a goal keyframe, and in the end the robot will have different size demonstration sets for learning the goal model and the action model. The intention with this alternative is to provide a wider variety of ways that the teacher can provide goal demonstrations to the robot.

The participant stands to the right of the robot during the kinesthetic demonstrations (see Fig. 32(a)), and is positioned on the opposite side of the table during goal-only demonstrations (see Fig. 32(b)).

The speech commands presented in Table 7 are used to switch between the demonstration modes.

7.1.3 Skills

The practice skill is *Touch*, with the goal of touching two objects in a given order. This skill is used to get the participants familiar with the two types of demonstration (kinesthetic



(a) A teacher providing a kinesthetic demonstration of a box closing skill to the robot.

(b) A teacher providing a goal-only demonstration of the pour skill to the robot.

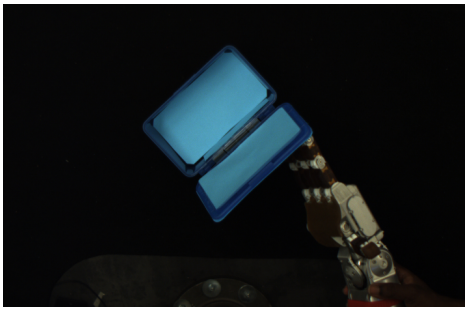
Figure 32: Types of demonstrations

and goal-only) and the keyframe interaction dialog in general. It is setup such that an intermediate keyframe is needed to move between two objects to highlight the keyframes. The two evaluation skills are as follows.

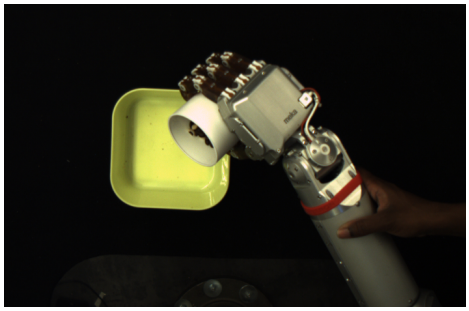
In the *close the box* skill, the aim is to close a particular box, as seen in Fig. 33(a). The reference point of this skill is the box. The pose of the end-effector is encoded with respect to the box reference frame. The centroid and angle of the bounding box features for the keyframe sequence is also encoded with respect to the reference frame of the box in the first keyframe. The success metric for this skill is whether or not the box lid is closed.

The aim of the *pour* skill is to pour coffee beans from the cup to the square bowl, as seen in Fig. 33(b). We assume the object of interest is the target bowl, since the cup can be considered as the part of the end-effector. The end-effector pose and the relevant features are encoded with respect to the bowl. The amount of coffee beans in the cup is measured to be the same at the start of the skill across all demonstrations.

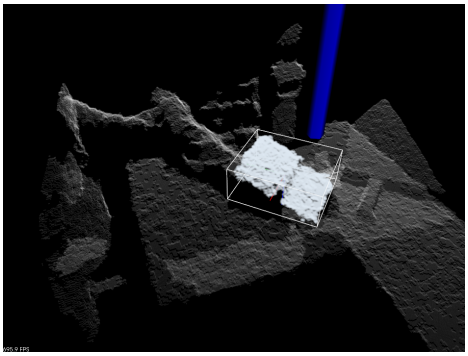
An interesting future work is to select the objects of interest and the reference points either automatically or through user interaction. However, for this experiment, they are



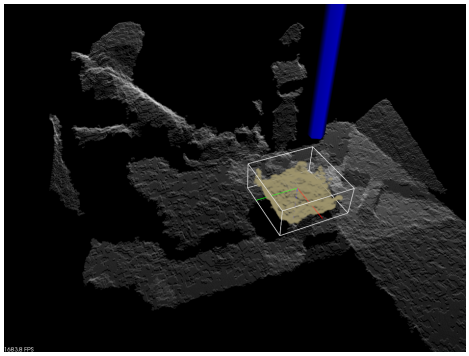
(a) A snapshot of a keyframe from the close the box skill.



(b) A snapshot of a keyframe from the pour skill.



(c) A segmented box for close the box skill.

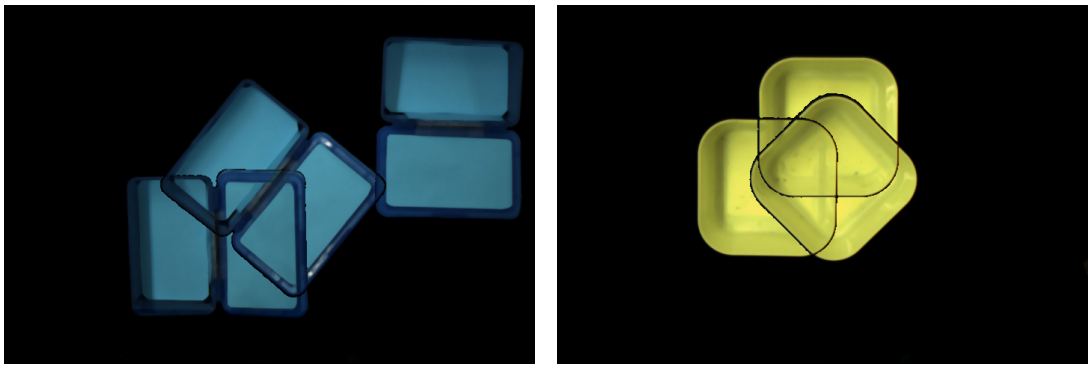


(d) A segmented bowl for the pour skill.

Figure 33: Image snapshots as seen by the overhead camera.

fixed, and this is the only skill specific representation decision that is made.

These two skills are very different from each other in terms of both object data and motion data. These skills are chosen in order to show that we can learn different goal models without engineering the feature space for a particular task. These are two different examples of the class of object directed motion tasks that the approach described in Chapter 6 is designed for. While the experiment would benefit from including even more skills, this decision is a trade-off with collecting a greater set of demonstrations from a single user. Instead each participant was asked to do six complete demonstrations and three goal-only demonstrations of each skill, which took around 30 minutes to complete. This was the target length for teaching sessions since longer sessions risk losing the participant's interest and could affect the quality of data, especially considering that this experiment collect batch data as opposed to being interactive.



(a) Close the box (b) Pour

Figure 34: The three poses of the objects for demonstrations for both skills overlaid.

7.1.4 Additional Details

The experiment had 8 participants (4 male and 4 female) with ages between 18-26 (median 21.5). They were recruited from the campus community, and none had prior experience interacting with a humanoid robot in an LfD setting.

For each skill, there are three initial poses of the reference objects. These poses can be seen in Fig. 34. For each pose of the object, the participant is first asked to show a goal-only demonstration and then to provide two kinesthetic demonstrations. The objects are placed such that the same point of view for the user is maintained for both of the demonstration modes[1]. The overall experiment results in 18 demonstrations $(2 \times (6+3))$ from each participant.

The reason to collect multiple demonstrations from different poses of the objects is to build a more general model of the action, as pointed out in Sec. 6. As a result, a direct playback of the user demonstrations is not always feasible to execute the skill. For example, the arm motion required for completing the close the box skill for the horizontal box position in Fig. 34 would be out of the robot's workspace if transformed for the vertical box position. Similarly for the pour skill, the demonstrations for the rotated bowl would not be applicable to the non-rotated bowl, unless prior knowledge of rotation independence

[1]They are mirrored since the participant is standing across the table in one and standing next to the robot in the other

of the skill is used.

7.2 Evaluation Overview

First evaluation for the models is done via cross-validation with the demonstration data. This analysis is done for both the action and the goal models in order to show the level of similarity of the demonstrations in the two different feature spaces. However, cross-validation is just a first step to assess model performance. The real aim of the goal models is to provide information about the success of the skill execution, and the aim of the action model is to produce successful executions of the skill. To this end, both models are evaluated in a series of robot trials. The action models are evaluated by running the generated trajectories on the robot and the goal models evaluated on recognition accuracy of the success/failure of these executed actions.

This analysis uses $log(\tau_s) = -500$ for the goal models and $log(\tau_s) = -1000$ for the action models. The threshold for the goal model is chosen such that there is correct classification of both successful and unsuccessful trials, based on cross-validation results. The threshold for the action models is chosen based on the distinct cutoff of likelihood estimations; anything below the selected threshold was too low (*e.g.* at the smallest floating-point value). 1 demonstration of participant 4 and 1 of participant 8 for the pour skill was thrown out due to the object being fully occluded in one of their keyframes.

7.3 Cross-Validation on Demonstration Data

7.3.1 Aggregate Models

The first analysis is designed to show the similarity between the participant demonstrations. A modified k-fold cross-validation is used for this purpose. Instead of randomly dividing the data, one participant's demonstration is left out as test data and a single aggregate model is trained with all of the other participants' demonstration data together. Since there is more goal data than action data, due to goal-only demonstrations, the same analysis is ran

twice for the goal models; once for all the demonstrations and once with only kinesthetic demonstrations (removing the goal-only demonstrations).

- Goal Model Recognition Accuracy: The average results for all the users is 100% correct recognition for both the close the box skill and the pour skill. This shows that the users' demonstration were overall similar to each others'. The result is the same, 100% correct recognition, for both of the skills when the goal-only demonstrations are removed.

- Action Model Recognition Accuracy: The average recognition accuracy of the action models across all users is also high, 89.6% for the close the box and 97.5% for the pour skill. These results suggest that the demonstrations are consistent overall with a relatively large set of data.

It should be noted that this analysis includes tests with only positive examples since there were no failed demonstrations.

7.3.2 Between Participants

Next evaluation looks at the generality of each individual participant's model with respect rest by training with a single participant's data and using the other seven participants' demonstrations as test data. This analyzes the ability of the model built from one participant's data to generalize to other participants' data. The action models performed very poorly in this task, even though they performed well with the aggregate data. As a result they are not included in this analysis.

These results are shown in Table 8(a). For the close the box skill, apart from participant 1, all the other participants had better than chance goal recognition performance and participants 2,4,5,6 and 7 had very good performance. Participant 1 has only provided between 2 or 3 keyframes per demonstration whereas other participants provided 4-6. As a result, participant 1s goal model was not able to recognize the demonstrations of

other users. The recognition performance for the pour skill is lower but all participants did better than chance with participants 5,6 and 7 had very good performance. These results imply that the participants provided perceptually similar demonstrations. As seen in the table, some participants (4,5,6) provided quite general demonstrations (*i.e.* , good variance) across both skills (higher than 80% accuracy); and the average recognition accuracy for all the participants was 74.2% for close the box, and 75.2% for pour. These results are quite good considering the low number of training data in this analysis (1 participant = 9 demonstrations) and the high dimensionality of the feature space of the goal model. The results without the goal-only demonstrations have slightly lower success rates apart from participants 1 and 4 for the pour skill. This is because the kinesthetic demonstrations tend to have have more occlusions due to having both the user's hand and the robot's end-effector over the target bowl in the frame of the sensor, resulting in worse data for the pour skill.

7.3.3 Within Participants

Lastly, the recognition performance of each participant's goal model is evaluated with their own demonstration set by applying leave-one-out cross-validation on this set. These results are seen in Table 8(b). The recognition results for the goal models are similar across both skills and as expected are better than the generalized 1vs7 task, with an average 93.6% accuracy for close the box and 91.0% for pour. The results without the goal-only demonstrations are very similar for the close the box skill but differ for some participants in the pour skill. The reason is same as before, more occlusions when both the users' hands and robot's end-effector occluding the object.

The action models were not successful in this cross-validation recognition task as well. This was somewhat expected in the between-participant case, due to a wide range of possibilities to demonstrate the skill and user differences. However, the within-participant results show that the number of demonstrations we obtained is not enough to model the variance and/or the different ways to execute the skill. This does not imply that good action models

Table 8: The cross-validation results for the goal model. Avg. refers to the average results. The columns under "All" refers to data including goal-only demonstrations and "Reduced" refers to data from only kinesthetic demonstrations.

(a) 1vs7: Trained with 1 user, tested against 7

	Close the Box		Pour	
	All	Reduced	All	Reduced
1	0	0	76.2	30.1
2	88.9	71.4	60.3	45.2
3	58.7	38.1	71.4	83.3
4	98.4	95.2	95.2	47.6
5	88.9	85.7	81.0	73.8
6	98.4	92.9	85.7	71.4
7	96.8	100.0	71.4	61.9
8	63.5	42.9	60.3	71.4
Avg.	74.2	65.8	75.2	60.7

(b) Single User: Cross-validation with the 8-9 demonstrations for a single user

	Close the Box		Pour	
	All	Reduced	All	Reduced
1	100.0	100.0	88.9	66.7
2	88.9	83.3	100.0	66.7
3	100.0	100.0	100.0	100.0
4	88.9	100.0	75.0	40.0
5	100.0	100.0	100.0	100.0
6	77.8	100.0	100.0	100.0
7	88.9	100.0	88.9	100.0
8	100.0	66.7	75.0	40.0
Avg.	93.6	93.8	91.0	76.7

cannot be learned but imply that the demonstrations are diverse.

7.3.4 Discussion of Cross Validation Results

These three sets of results show that demonstrations are similar and consistent both between and within the users in the goal space, and even 9 demonstrations per user is enough to learn accurate goal models for these skills. High recognition rates without the goal-only demonstrations (*i.e.* 6 demonstrations) are also shown. However, more action data was needed to span the state space with varied examples. These results are expected given that (1) naïve users are goal oriented and (2) there are many ways to accomplish the same skill in the action space but all of these will look similar in the goal space.

7.4 Robot Trials: Skill Execution

One aim of learning both models is to be able to execute the skill on the robot and know whether the execution succeeded or not. This is arguably the main purpose of LfD, making use of the learned models in practice. The individual learned action models had very low recognition performance in the cross-validation tests (both within- and between-participants), but low cross-validation scores do not imply that the resulting action models are useless, a more fair analysis of the action models is evaluating their success at generating motion. The cross-validation tests how similar the demonstrations are but not how the action itself is modelled. This analysis is performed by executing the skills with each of the learned action models, using Alg. 1. An example of executing a learned action model for the close the box skill is depicted in Fig. 35.

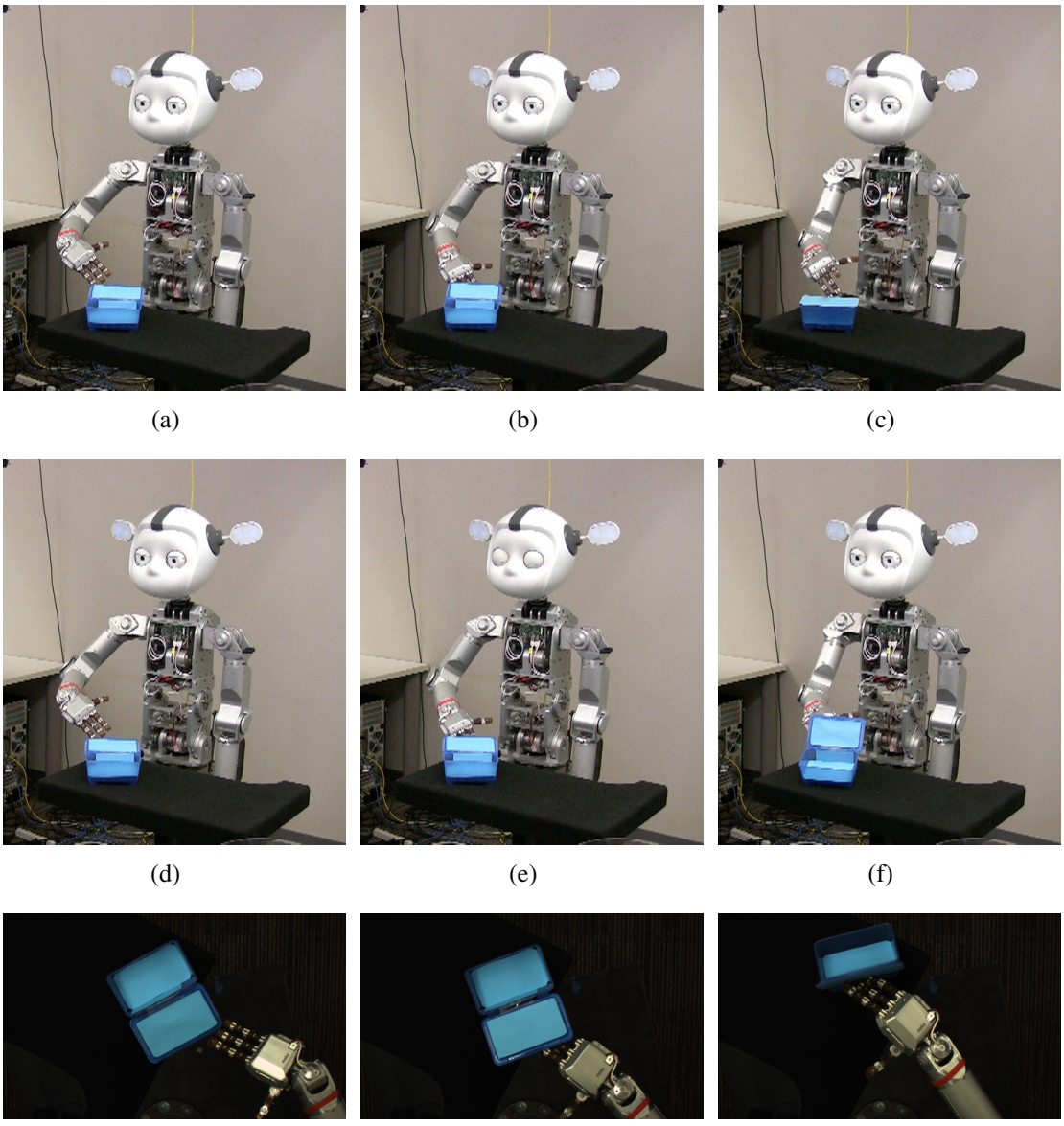
The learned action models are tested for each skill for the 8 individual participants. An action model of a user was executed 5 times and the success or failure results were noted as the ground truth. The fully closed box was regarded as successful for the box skill. The pour skill was regarded successful if the robot was able to pour most of the coffee beans to the bowl (i.e., a bean or two bouncing out was still called success). These results are seen in Table 9, under the *Execution Success* columns. There is a wide range of success rates

across participants. Six of the eight participants achieved a 100% success rate (5/5) with at least one of their skills, and two people did so for both. Whereas three people had one of their action models with a 0% success rate. In general the pour skill was more successful, with a 75% success rate across all participants, compared to the close the box skill with a 57.5% aggregate success rate.

There are a few common modes of failure. For the close the box skill, the fingers sometimes touch too lightly to the lid and lost contact. This is exacerbated by the highly complaint fingers of the robot as they bend slightly with touch. Another case is when the fingers get stuck on the body of the box and tilt it instead of closing it, as shown in the bottom row of Fig. 35. This happens due to user demonstrations; not enough clearance is demonstrated when going from under the lid to over the box. In an interactive scenario, the teacher might realize this and fix it with their follow-up demonstrations. For the pour case, the common mode of failure is having not enough downward rotation to pour the entire cup.

7.5 Robot Trials: Skill Monitoring

While executing the action models, the robot extracts an observation frame in the goal space at each action keyframe of the skill, as described in Sec. 6.4.2, and forms an observation sequence. The sequence is then input to the goal model for the corresponding participant to calculate a likelihood which is then thresholded to decide on success or failure of the execution. These results are seen in Table 9, under the *Monitoring Results* columns. This table shows the correctly recognized execution outcomes (true positives and true negatives) and the mistakes (false positives and false negatives) for each participant's goal models across both skills. The recognition accuracy at monitoring is good for both of the skills. Looking at the overall rates, only 4 of the the 40 trials was incorrectly classified across both skills. Thus, a 90% success rate in the goal monitoring task for each skill. The interesting result is that even for participants with low execution success, their goal model is good.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 35: Image snapshots for a close the box execution. The first row shows a successful execution and the second row shows a failed one. In the failure case, the robot's fingers got stuck to the body of the box during, as shown in the last row. As a result it tilted the box, instead of closing it.

Table 9: Skill Execution and Monitoring Results

	Close the Box			Pour		
	Execution	Monitoring Results		Execution	Monitoring Results	
		True	False		True	False
		Pos : Neg	Pos : Neg		Pos : Neg	Pos : Neg
1	100%	5 : 0	0 : 0	100%	5 : 0	0 : 0
2	0%	0 : 5	0 : 0	100%	5 : 0	0 : 0
3	60%	2 : 2	0 : 1	0%	0 : 3	2 : 0
4	40%	2 : 3	0 : 0	100%	4 : 0	0 : 1
5	40%	2 : 1	2 : 0	100%	5 : 0	0 : 0
6	100%	5 : 0	0 : 0	100%	5 : 0	0 : 0
7	40%	2 : 2	1 : 0	0%	0 : 4	1 : 0
8	80%	4 : 1	0 : 0	100%	5 : 0	0 : 0
Average	57.5%	22 : 14	3 : 1	75%	27 : 9	3 : 1

This is an expected result based on the successful cross-validation results, but is reassuring to see that the goal models perform well on new data observed when the robot is executing the learned action models.

7.6 Summary

This chapter presents results of evaluating the action and goal learning approach described in Chapter 6. Towards this end, data from eight naïve users for 2 skills were collected It was seen that the skill demonstrations are more consistent in the goal space, both across users and within users. This confirms the observation about the goal-oriented nature of naïve users. Some users were not able to teach successful action models with the average success rates being 57.5% for the close the box skill and 75% for the pour skill. Successful goal models can be learned from all the users, even for users with less successful/failed action models. The average execution monitoring success rate was 90%.

The main take away form this experiment is that successful goal models can be learned from naive teacher demonstrations even if their action models are not as successful. This is an important step for robots to act autonomously in their environments. The robot can re-attempt the skill or call for help if it fails and move on to its next skill if it succeeds.

The results presented in this chapter sets up the self-improvement work presented in

Chapter 8, which uses the high monitoring success of the goal models to improve the action models. Self-improvement shows that the performance of the unsuccessful action models can be increased. This is done by sampling from the action model, executing the samples on the robot and using the monitoring output of the goal models for these executions to update the action models.

CHAPTER VIII

SELF-IMPROVEMENT

The work presented in Chapter 7 showed that the goal models are able to correctly label executions, as either success or failure, even in the case of underperforming action models. There are cases where end-users are not able to teach acceptable skills to the robot. In these cases, an extra self-improvement step is needed to have an acceptable model of the skill. This chapter builds atop this monitoring performance and introduces a novel method for using these learned goal models to guide self-improvement of the action model. This algorithm is called the *Goal based Learning and Exploration - GoaL-E*. The addition to the system is highlighted in Fig. 36. The work presented in this paper is published in [8].

The approach starts by learning action and goal models from demonstration as described in Chapter 6. After this initial phase of learning, the self-improvement begins. The idea is to sample from the action model, execute this on the robot and do goal monitoring. Then the action model is updated based on the goal model output. The main assumption is that there is a successful goal model available after the initial user demonstrations.

8.1 Algorithm

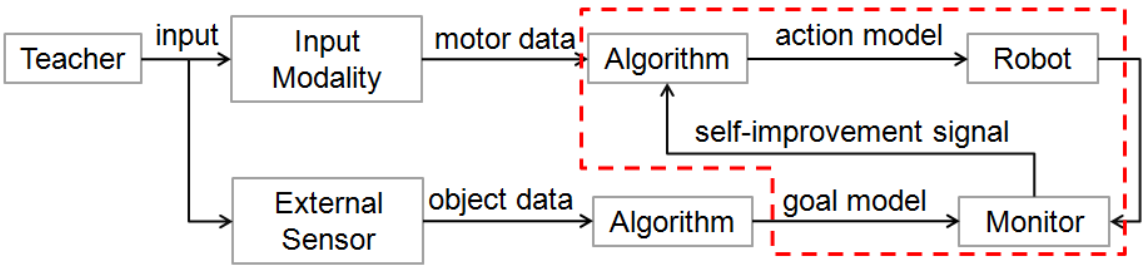


Figure 36: The augmented version of the LfD system in Fig. 30 that includes that includes the GoaL-E. The self-learning component highlighted with the red box. The robot executes the skill with variety and updates its action model based on the feedback from the goal model monitoring.

Algorithm 2 Self-Improvement(θ_G, θ_A, D_A)

1: $\sigma \leftarrow D_A$
2: $G \leftarrow [0, 1, \ldots, 0, 1]$
3: **for** 1 **to** n_e **do**
4: **for** 1 **to** n_r **do**
5: $r = \Sigma_{i=1}^{w}(G[(end - w + i) : end])/w$
6: $\lambda = f(r, h, \alpha)$
7: $T = sample(\theta_A, \lambda)$
8: $Q = execute(T, \rho_{obj})$
9: $g = monitor(\theta_G, Q)$
10: $G \leftarrow g$
11: **if** $g == 1$ **then**
12: $\sigma \leftarrow T$
13: **end if**
14: **end for**
15: **if** $forgetUserData(G, k)$ **then**
16: $\sigma = \sigma \setminus D_A$
17: **end if**
18: $\theta_A = learn(\sigma)$
19: **end for**

The self-improvement algorithm is presented in Alg. 2. It is an iterative algorithm that takes the user action demonstrations D_A, and the HMM models, θ_A and θ_G as inputs. For a given iteration, the robot samples from the action HMM (θ_A), executes the obtained trajectory and monitors it (lines 7-9). The *sample* step samples from the emission probabilities of the most likely state path instead of only taking the means, The execution and monitoring is as described in Chapter 6. If the execution is deemed successful, the sample is added to the set of successful examples, σ (lines 11-13) and the monitoring result is stored in G (line 10). After a number of iterations (n_r), the robot re-learns the action model using σ as described in Sec. 6.3. This overall process, which is called an *episode*, is repeated for a predetermined number of times (n_e) but a stopping condition can be used as well.

Successful sampled trajectories are more relevant to learning the skill than user demonstrations since they are executed by the robot and user demonstrations can potentially be

bad. The effective stiffness of the robot's arm is different between kinesthetic demonstrations (*i.e.* when the user is holding the arm) and when the robot is executing the skill. Moreover, kinesthetic teaching only ensures that the demonstrations are within the workspace of the robot and doesn't guarantee dynamic concerns (*e.g.* whether the robot can move fast enough). Even when these are not issues, the teacher demonstrations might not be sufficient to learn a good model. Hence, the user demonstrations are "forgotten" if there are sufficient successful samples (lines 15-17).

There are two versions of the *sample* of this algorithm (line 7); *normal* and *adaptive*. In the normal case, the samples are drawn directly from the emission distributions. In the adaptive case, the covariance matrices of the emissions are multiplied by a scalar factor is which is calculated according to the success of the last w samples (lines 5-6). This is introduced in the next section.

8.2 Adaptive Sampling

The self-improvement method is essentially a search guided by the goal model. The adaptive sampling is introduced to adjust this search. It is assumed that the best opportunity for learning is on the border of success and failure, *i.e.* point of maximum entropy. In other words, the method tries to fail and succeed the same number of times during the search to maximize the information gain. One assumption is that the learned action model is either within the boundary of success or close enough to the boundary to be found. This is similar to the assumption of the initial model being within the basin of attraction of a successful local minima in policy search methods.

A multiplication factor for the covariance matrices, λ, is used in the *sample* step of Alg. 2. Instead of sampling from $\mathcal{N}(\mu,\Sigma)$, the method samples from $\mathcal{N}(\mu,\lambda\Sigma)$ to get execution keyframes, where $\lambda \geq 1$. Increasing the covariance like this makes it more likely to sample away from the mean. The Fig. 37 provides a 1-dimensional depiction. This allows the method to scale its search to be between the vicinity of the current model and

farther out.

Eq. 1 shows the calculation of the step size parameter (line 6).The r represents the success ratio of the last w samples as calculated at line 5 of the Alg. 2, hence $0 \leq r \leq 1$. Note that for $r = 0$, the equation is undefined but as $r \to 0$, $f(r,h,\alpha) \to 1+\alpha$. The G parameter is initialized (line 2) with a set of equal number of $1's$ and $0's$ to avoid r being over-sensitive to initial sampling results. The α parameter is responsible for the maximum step size and h is responsible for the width of the function. The output of this function for a few example parameters is shown in Fig. 38.

$$f(r,h,\alpha) = 1+\alpha\left(1-exp\left(-\frac{log^2(\frac{1-r}{r})}{h}\right)\right) \tag{1}$$

For $0 \leq r \leq 1$, the Eq. 1 is symmetric, non-negative and $1 \leq f(r,h,\alpha) \leq 1+\alpha$. It reaches its minimum at $r = 0.5$ and maximum at $r = 0$ and $r = 1$. These properties result in the self-improvement method to look farther out if the latest sampling results are similar and to stay within the vicinity of the current action model when the sampling results are different. This forces the algorithm to spend more time close to the success/failure boundary, as previously motivated.

8.3 Evaluation

The robot used in this study is Curi and the experimental setup can be seen in Fig. 40 (a-b) and Fig. 39. A RGBD camera overhead is mounted above the table. Both simulation and real robot experiments are used to evaluate the self-improvement approach. The evaluation starts by demonstrating skills to the robot and learning goal and action models. Then, these models are input to the self-improvement algorithm where the action models are updated. In both cases, the real robot is used to provide demonstrations and the simulated robot is programmed to mimic the real one.

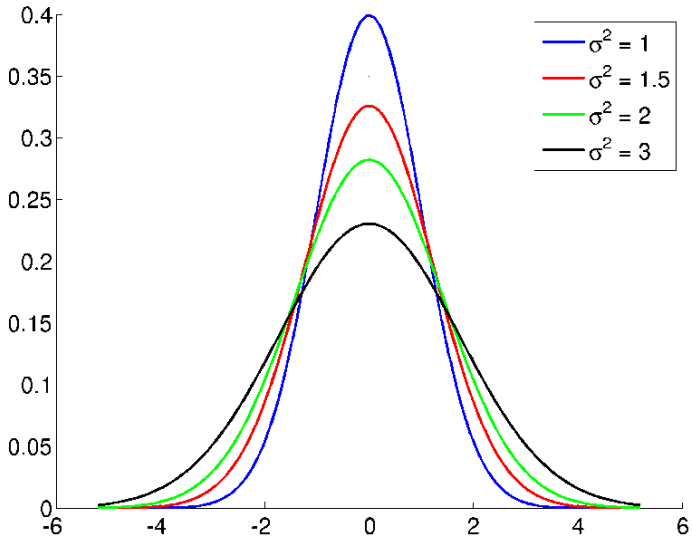


Figure 37: The 1-dimensional depiction of the effects of multiplying the variance of a Gaussian distribution by a scalar factor that is larger than 1. The distribution gets flatter and as a result, the probability from sampling away from the mean increases.

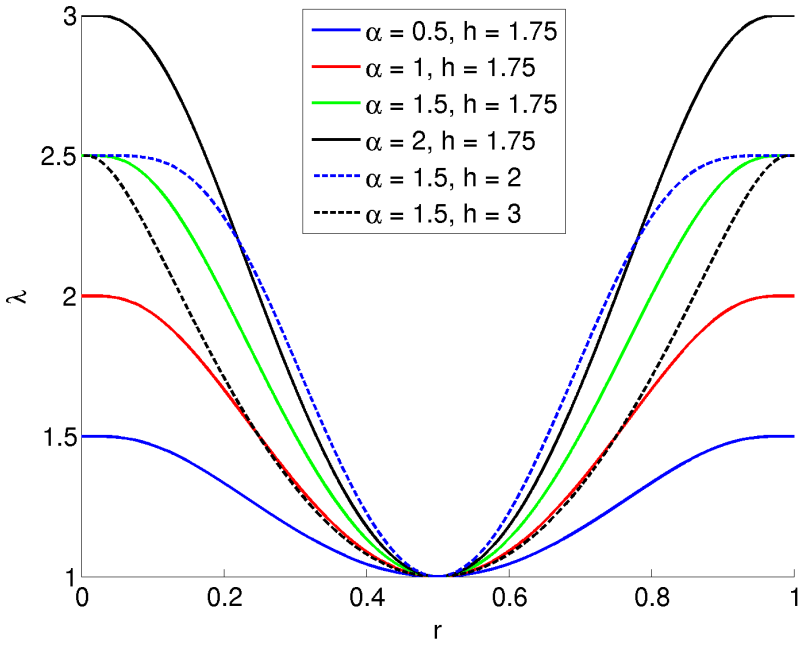


Figure 38: The step-size parameter (λ) versus sampling success ratio for various values of α and h as calculated by the Eq. 1.

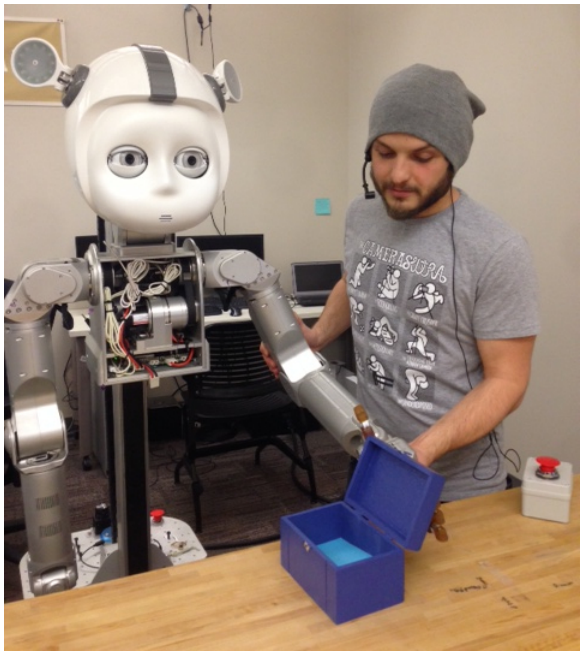
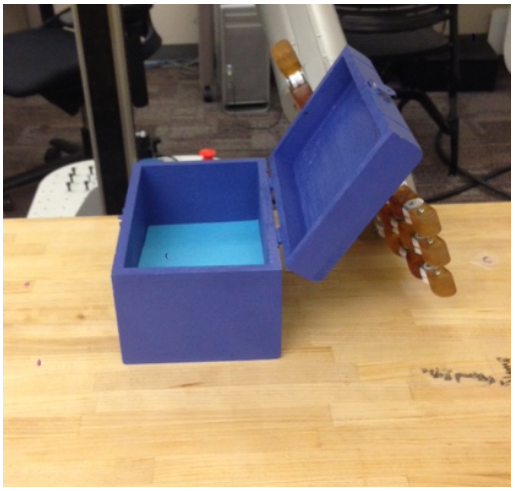


Figure 39: A teacher providing a kinesthetic demonstration of close the box skill to the robot.

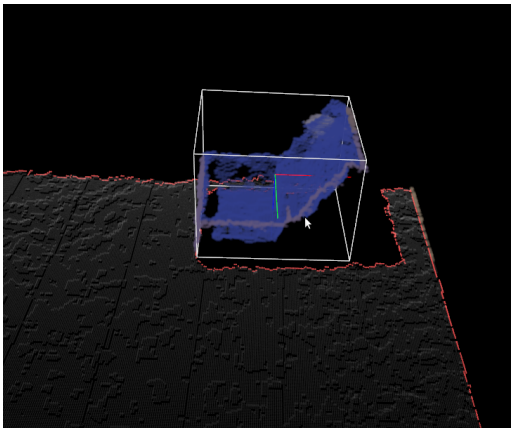
8.3.1 Feature Space for Goal Models

An example of the segmentation and the bounding box results can be seen in Fig. 40. The general version of the features extracted from the segmented objects was introduced in Sec. 6.2.1. The specific features for this evaluation include the bounding box coordinates (3) and orientation (1), cluster centroid (3), minimums and maximums of the point cloud coordinates (6), average RGB values (3), average hue (1), point cloud size (1), bounding box size (3), volume (1), area(1), aspect ratio(1), bounding box area to volume ratio (1) and bounding box volume to point cloud size ratio (1). The color values are mapped between 0 and 1. Regarding the VFH features, 15 bins per angle is used. In the end resulting goal space is 71 ($26 + 15 \times 3$) dimensional, treated as $\mathbb{R}^{71}$. An expert did not fix any of the clustering issues as was done in Experiment III (see Sec. 7.1.1).

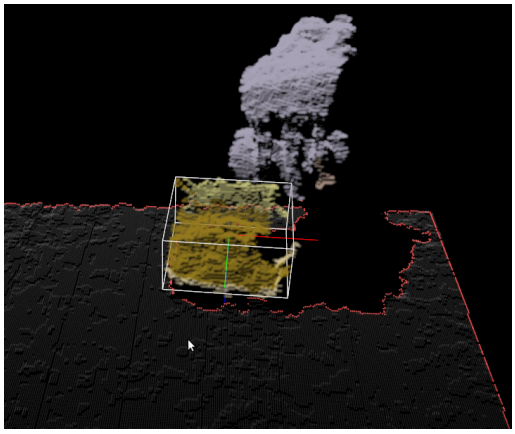


(a) A snapshot of a keyframe from the close the box skill.

(b) A snapshot of a keyframe from the pour skill.

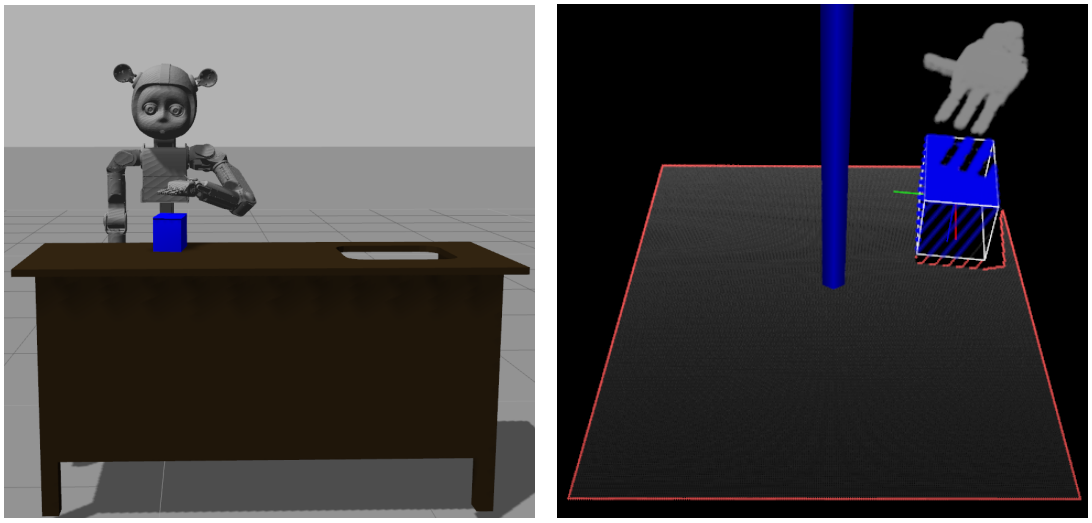


(c) A segmented box for close the box skill.



(d) A segmented bowl for the pour skill.

Figure 40: Image snapshots as seen by the overhead camera.



(a) The robot in simulation (b) The perception in simulation

Figure 41: The simulated environment.

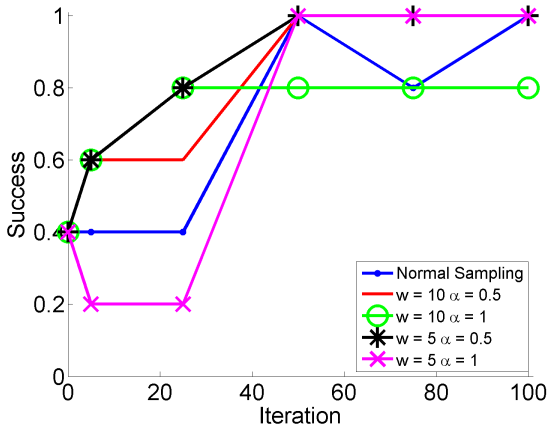
8.3.2 Simulation Results

The simulator used is Gazebo 4.0[1]. Screenshots from the simulation and object segmentation from simulated data can be seen in Fig. 41. The *close the box* (CLB) skill is used to evaluate the approach in simulation; in which the goal is to close the lid of an open box.

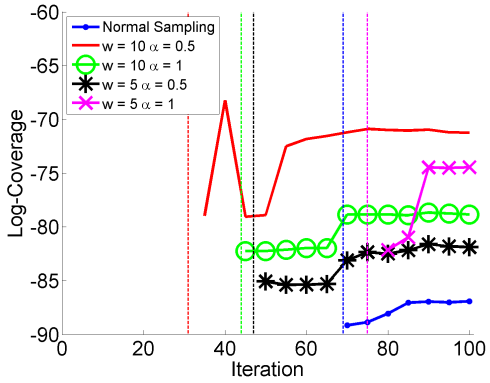
The method is tested with two initial action models, one successful (success rate 100%) and one unsuccessful (success rate 40%). Rather than purposely providing bad demonstrations, failure data is generated by modifying good demonstrations; by adding a constant bias of $0.027m$ to the vertical and horizontal dimensions of the second keyframe, forcing it away from the box. For reference, the box dimensions are $0.165m \times 0.108m \times 0.103m$. The unsuccessful action model is then learned from this modified data. The goal models are shared for both action models.

The approach is tested under multiple parameter instantiations of the Eq. 1. The width parameter is fixed at $h = 1.75$. The success ratio window size, w and the maximum step size, α are varied. The list of parameters we use is $[\alpha = \{0.5, 1\}, w = \{5, 10\}]$,. The teacher data is discarded after getting $k = 10$ successful samples (Alg. 2, line 16). The the

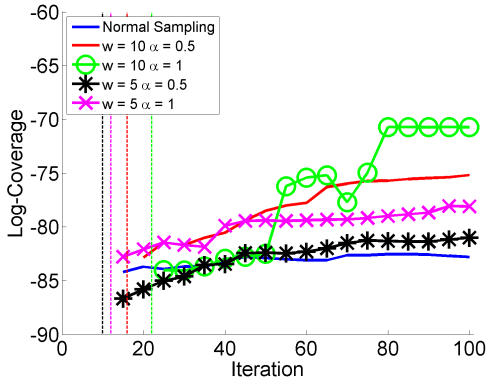
[1] `http://gazebosim.org/`



(a) Success rates for episodes $\{1,5,10,15,20\}$ for the Unsuccessful Initial Model



(b) Coverage for the Unsuccessful Initial Model



(c) Coverage for the Successful Initial Model

Figure 42: Simulation: The success rates and the coverage of the action models versus iterations of the self-improvement algorithm for the close the box skill. The vertical dashed lines represent the point of forgetting the user data.

algorithm is run for $n_e = 20$ episodes with $n_r = 5$ iterations each. The goal model log-likelihood decision threshold is set at $\tau_s = -600$. In addition, the algorithm is ran without adaptive sampling and compared against the adaptive sampling version.

The results for the close the box skill is given in Fig. 42. After each episode, a new HMM is learned. The Fig. 42(a) shows the average success over 5 executions for a selected set of learned HMMs. Eventually, all the instances of the algorithm reach a successful ($\geq 80\%$) state within 10 episodes.

The interesting result is that the case without adaptive sampling (normal sampling) also managed to improve the skill model. The results for the successful initial model is not

shown, since, they are all 100%. One assumption of our method is that the initial action model is not too far from the achieving the skill (see Sec. 8.2). In this case, successful skill executions were within the variance of the initial unsuccessful goal model and hence were able to recover an acceptable action model.The adaptive sampling is expected to have more impact when the initial model is farther away, which is actually the case for robot trials.

The volume of the emission probabilities is an indicator of the state space coverage for an action model. The volume of hyper-ellipsoids represented by the covariance matrices is used and the *coverage* of an action model is defined as the sum of the determinants of these emission covariance matrices. The Fig. 42(b) and Fig. 42(c) shows the coverage of the learned models after the user data is forgotten, as the user data for the unsuccessful demonstrations is un-purpose different than successful sampled trajectories and skews the results. The figures show that the adaptive sampling is faster in increasing the coverage and as a result faster at searching the state space.

8.3.3 Robot Results

The method is evaluated on the real robot using two skills: the close the box (see Fig. 40(a)) skill, similar to the simulation case and the pour (see Fig. 40(b)) skill, where the goal is to pour uncooked macaroni from a cup to a bowl. The algorithm is run for $n_e = 10$ episodes with $n_r = 5$ iterations each. The parameters of Eq. 1 are fixed at $[\alpha = 0.5, w = 10, h = 1.75]$ based on the simulation results as these parameters resulted in a high success rate and good coverage. The user data is discarded after getting $k = 10$ successful samples. The results for already successful skills are not expected to be different than the simulation case and as such only the success rates starting from initial unsuccessful models are presented.

The author teaches a successful goal model using 10 demonstrations for both skills but start the self-improvement with unsuccessful action models. For close the box skill, the demonstration data is modified to result in an unsuccessful action model, similar to the simulation case. The second keyframes are pushed away by $0.03m$ in horizontal and

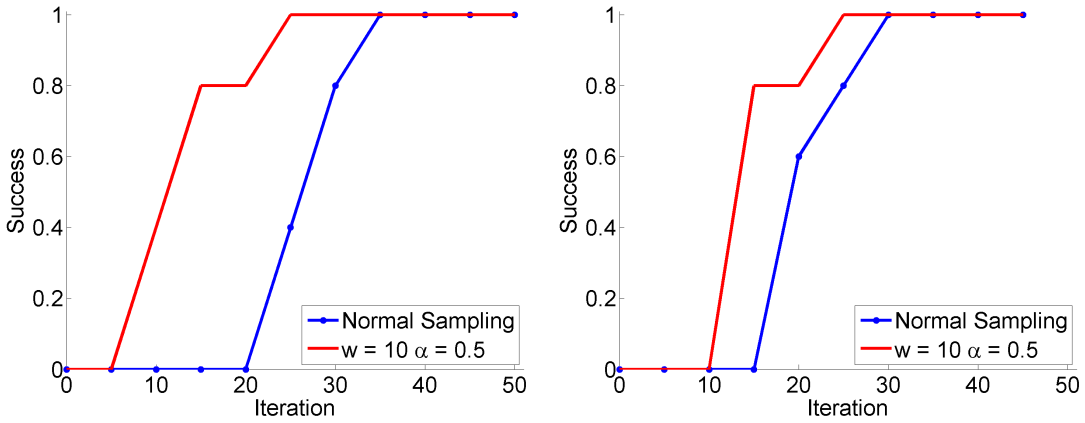
vertical directions away from the robot and the third keyframes are pushed away from the box by $0.03m$ in the vertical direction. For the pour skill, we provide 10 bad demonstrations to the robot. The resulting unsuccessful action models has the cup 1-2 cm away from the bowl and is not tilted enough to pour the macaroni pieces. Both of the initial action models had 0% success rate. To evaluate the approach, the action models are executed 5 times after each episode. The resulting success rates are shown in Fig. 43.

For close the box skill, there are differences between the simulation and robot experiments. The robot experiment starts from a completely failed model (success rate 0%), whereas simulation experiment starts from a partially failed model (success rate 40%). Both of the robot experiment conditions reached a successful action model faster, despite the fact that they start from a worse one. There are two main reasons for this. The first is that the covariance of the robot action models are higher, which results in a larger search space. The other is that it is actually easier for the real robot to close the box, given that the robot is compliant and has soft fingers, which lets it better interact with objects.

The adaptive sampling reached a successful action model faster. The reason being that the initial action model was farther away from the successful samples than in the simulation experiment, as evidenced by the initial success rate of the model. In this case, adaptive sampling was able to sample successful executions faster than the non-adaptive case. Adaptive sampling takes better advantage of a wider variance than the non-adaptive case; the more variance the model has, the more the adaptive step will grow it.

The results for the pour skill is closer together (Fig. 43(b). Both the adaptive and the non-adaptive methods were able to improve the action model and did so after 3 and 4 episodes respectively. As expected, the adaptive sampling was slightly faster at improving the skill.

The success of the goal models is also evalauted since the approach depends on successful goal models The sampling and evaluation of the skill resulted in 100 iterations of monitoring for each skill. The close the box goal model had 93% recognition rate and the



(a) Close the box (b) Pour

Figure 43: Real robot: The success rates for the close the box and the pour skill after each episode for 5 trials.

pour goal model had 91% recognition rate which is inline with the naïve user experiments presented in Chapter 7.

8.4 Summary

This chapter introduced a novel approach to self-improvement of skills learned from demonstration. This approach builds on observations of naïve users being goal oriented. Goal and action models are learned using demonstration data. Then, goal models are used in a self-exploratory way to improve the action models without further user interaction. An adaptive sampling method is introduced to handle the exploration versus exploitation trade off.

The main contribution of this chapter for the purposes of this thesis is to provide an approach to improve learned action models using learned goal models. Prior to the work described in this chapter, there was no LfD approach that can learn skill models for high dimensional robots, both action and goal wise, and improve them without further user interaction, programming and/or heavy prior knowledge for object centric manipulation skills.

This chapter evaluated self-improvement with expert data. The Chapter 9 evaluates self-improvement within an interactive LfD system with non-expert data.

CHAPTER IX

INTERACTIVE LEARNING

This chapter adds the interactive component to complete the final version of the learning from demonstration approach, depicted in Fig. 44. An interactive LfD approach has the potential to increase the learning the performance over a batch approach. The experiments presented in Chapter 4 used interactive learning. However, the rest of the thesis work concentrated on developing learning methods and either worked with an expert user or with batch data. The final version of the LfD approach developed in this thesus work, depicted in Fig. 44, brings back the interactive component and closes the loop to create the *Interactive Goal based Learning and Exploration - iGoaL-E*. iGoaL-E combines the interactive system depicted in Fig. 5 and the system with self-improvement in Fig. 36 with additions for further interaction.

9.1 Description of iGoaL-E

The simplified version of the interaction flow of iGoaL-E is illustrated in Fig. 45. The teaching interaction starts with the robot waiting on input from the teacher, which is called the *IDLE* state. Then, the teacher can provide a demonstration to the robot in the *DEMO* state. Then the robot stores both motor and object data and learns action and goal models from this data in the *LEARNING* state, with the algorithms described in Sec. 6.3.

The interaction opportunities that the iGoaL-E affords the teachers include the ability to have the robot execute its current action model during teaching. This is done in the *EXECUTION* state by using the action model and the method described in Sec. 6.4.1. Executing the learned model is an implicit way to communicate the state of the learned model. The assumption is that seeing this execution, the teacher will have a better understanding of the state of the learned models and tailor the next demonstrations accordingly which

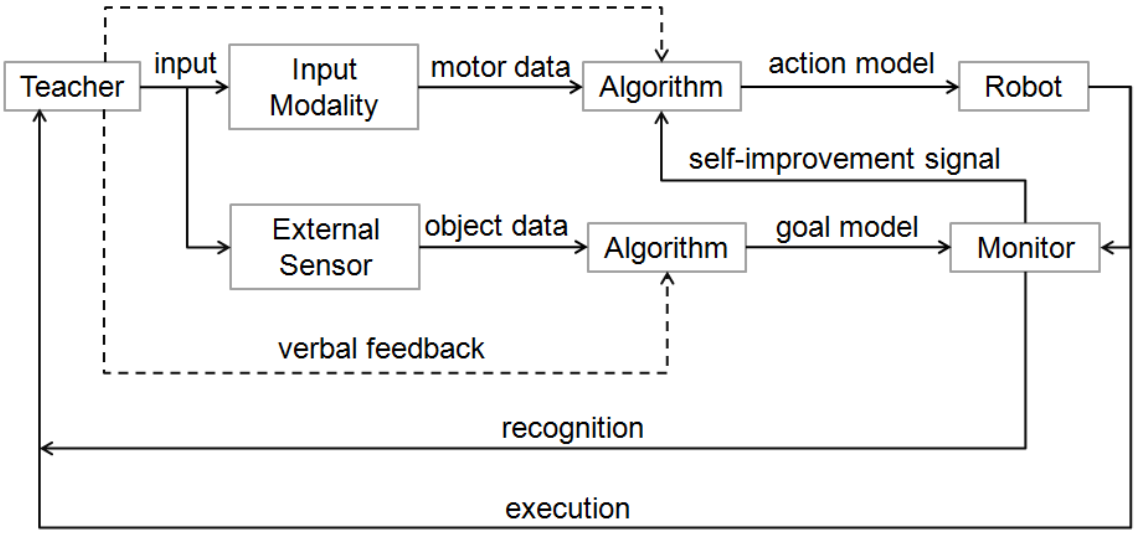


Figure 44: The final version of the interactive LfD system developed in this thesis. The teacher is able to see the robot's executions and hear the robot's recognition guess. These communicate the state of learning for the action and the goal models respectively. This in turn can influence the teacher to tailor his/her demonstrations for a higher learning performance. In addition, the user can provide verbal feedback after robot's execution, both during teaching and sampling phases, which provides additional data to update the learned models.

will result in better learning performance. If the robot does not learn immediately after the demonstrations and/or execute it immediately, the teaching becomes batch data collection, which was the case for Experiment III.

The execution of the skills during the teaching interaction presents an opportunity to get more data for updating goal models. As the robot executes the skill, it also collects data for monitoring (see Sec. 6.4.2). At the end of an execution, the robot asks the user how it performed. At this point, the user can give binary feedback as either success or failure. If the user declares success, then the robot uses stores this goal data and updates its goal model (switching from the EXECUTE state to LEARNING state). The robot discards the data otherwise. The reason that the action model is not updated if the robot is executing the default trajectory is to prohibit the model to bias itself towards this trajectory as calculated in Sec. 6.4.1. In addition to executing the action, the robot also verbalizes its monitoring result at the end. This is useful for telling the user about the state of the goal model.

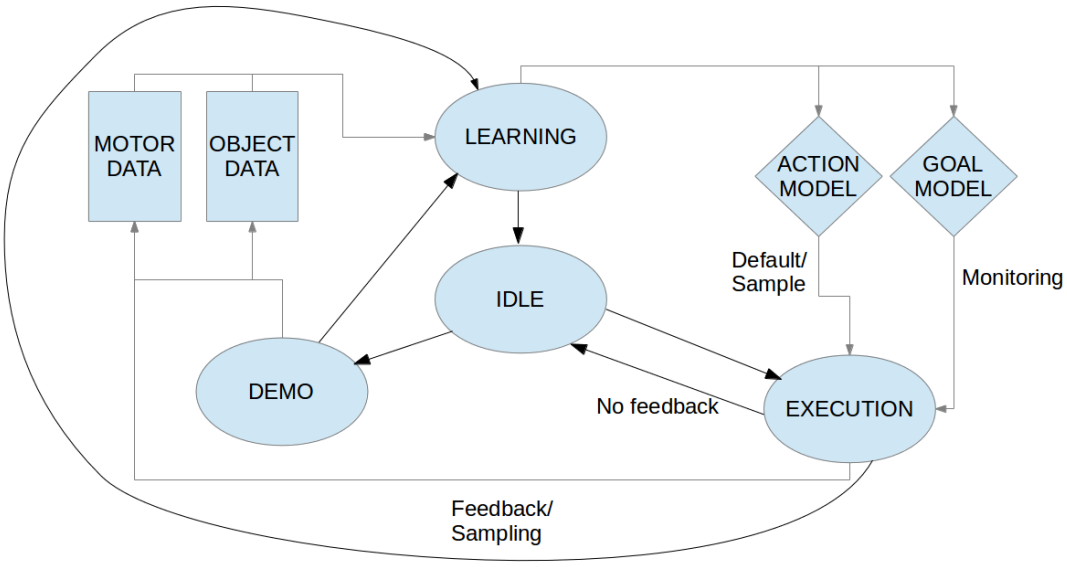


Figure 45: The ellipses represents the states of the interaction. The dark black lines depict the transitions between these states. The robot starts at the IDLE state. The teacher can provide demonstrations in the DEMO state, after which motor and object data are stored. At the following LEARNING state the robot learns an action and a goal model. The teacher can have the robot execute the default path or a sampled path coming from the action model which transitions the robot into the EXECUTION state. The robot verbalizes its monitoring output after executing the action. The teacher can give feedback on this execution by stating that it was either successful or not. During self-improvement, the robot executes sampled trajectories but uses its own monitoring output as feedback. The vertical rectangles represent the data stored during the interaction. This data comes either from the teacher demonstrations, teacher feedback on executions (the default path or sampled paths) or from self-improvement. The rotated squares represent the learned models.

The method, iGoaL-E, functions in two phases called the *teaching phase* and the *sampling phase*. Up to now, the interaction was in the teaching phase; the teacher gave demonstrations and asked the robot to execute what it has learned. iGoaL-E also lets the teacher be present during the sampling phase, when the robot does self-learning, and allows the user to provide verbal feedback for the sampled executions. Similar to the teaching phase, the robot asks for feedback after it samples and executes the skill. In the sampling phase, the algorithm introduced in Chapter 8, with some modifications (see Sec. 9.2), is used in the EXECUTION state to create sampled executions. If the user feedback is available, this is used to update both of the models, if it is not, the monitoring output of the goal model is used and only the action model is updated. The particular uses of the teacher feedback in this chapter is novel in the field of learning from demonstration.

In addition to resulting in reliable data, the user feedback is useful during this phase since the robot is more likely to come across object states it has not seen before. An example for this situation is illustrated in Fig. 46 where the robot's finger got stuck while opening the box. The robot did not see this during teacher demonstrations. User feedback can be invaluable in such cases if the previously unseen states are actually successes. Before asking for feedback, the robot has the option to communicate its recognition guess after the execution through synthetic speech. This is done to implicitly communicate the state of the goal model learning to the user.

After the interactive teaching and sampling phases, the robot can continue with the self-improvement phase. Moreover, the teacher can decide to move between phases whenever he/she wants.

9.2 Modification to Action Execution and Sampling

This section describes the modifications on the action execution (see Sec. 6.4.1) and sampling (see Chapter 8). These changes are motivated by a pilot study, not presented in this

Figure 46: The robot at the end of a sampled trajectory for open the box skill. The robot's finger got stuck at the lid of the box. The robot has not seen this during the demonstration phase

thesis. This was an open-ended study to analyze the verbal commands that naïve teachers preferred to use during learning from demonstration. The system depicted in Fig. 44 was used as the underlying LfD methodology. This study yielded that some action models learned from naïve users had most likely state paths (Sec. 6.4.1) that were too short. This was due to users providing keyframes at similar spots but at different parts of their demonstrations which causes the transition probabilities to be skewed. To overcome this, a prior on state paths is added to calculate state path likelihoods, which favors state paths close to the average number of keyframes provided.

In addition, the sampling approach described in Chapter 8 uses the most likely state path for all the samples. However, naïve users tend to change strategies either to fix the action or to provide variety, resulting in a branching structure for the learned HMM. By only using the most likely state path, the robot was losing opportunity to learn a more general action model. In addition, users found the sampling with the same underlying state path monotonic. As a result, the self-improvement process in this experiment also samples

from all the state paths between the prior states and the terminal states, based on their likelihood.

9.3 Experiment IV: Interactive Learning with People

Chapter 8 tested the self-improvement with an expert user. This chapter tests the final version of the interactive LfD system with naïve teachers, including an interactive improvement phase.

9.3.1 Experiment Details

The experimental setup is very similar to the one described in Chapter 8. The feature space for the goal models is the same as well. The number of skills is increased from two to three: The *open the box* skill is added in addition to the *close the box* and *pour* skills. The open the box skill was added since it is harder to teach a successful action model for it than the other two. This skill was added to test if the self-improvement would be able to fix a harder skill. The macaroni for the pour skill has been replaced by penne to facilitate easier cleanup if robot fails to pour. The skill success definitions are the same as before. For the open the box skill, the skill is considered to be successful if the robot was able to rotate the lid by at least 75 degrees.

The experiment had 12 participants (8 male and 4 female) with ages between 21-35 (median 25). They were recruited from the campus community, and none had prior experience interacting with a humanoid robot in an LfD setting[1].

The experiment followed the protocol described in Chapter 3 with a few modifications. In order to have a uniform experiment across participants, the interaction had the following structure. The participants were asked provide 5 demonstrations per skill to the robot. The robot executed its learned action model after 1^{st}, 3^{rd} and 5^{th} demonstrations. The reason for not executing after each demonstration was to limit the overall interaction to 60

[1]The participants were compensated with $10 for their time

Table 10: Additional Speech commands for evaluating iGoaL-E

Command	Function
Switch to Exploration	Move the interaction to the exploration phase
Try it yourself/Try again	Sample and execute in the exploration phase
That was a success/You succeeded	Give positive feedback
That was a fail/You failed	Give negative feedback
Yes it was/Yes you did	Agree with the robot's monitoring result

minutes. After each execution, the robot asked for feedback on its execution. The robot only updates its goal model using this feedback. The additional speech commands used in this experiment is given in Table 10.

In the thesis work of Cakmak ([17], heuristics provided to users were found to increase successful teaching.Motivated by this result, the users were given 3 suggestions to provide demonstrations: (1) try to stay as close to the object as possible, (2) try not to occlude the object too much, especially at the start and end of the skill, and (3) try to provide minimum number of keyframes that would still achieve the skill. The participants were also told that they do not have to follow these suggestions. These suggestions are based on the experience of the author.

After the teaching phase, the robot moved on to the sampling phase. It did five executions with sampling as described in Sec. 9.2. After each of these executions, the robot verbalized its recognition result and asked for feedback. Both of the models are updated after each execution based on this feedback. The number of demonstrations, executions and sampling were decided to limit the study to under 1 hour based on pilot testing. The participants are told that the teaching phase is called the *demonstration* phase and the sampling phase is called *exploration* phase.

9.3.2 Metrics

The action model execution and goal model monitoring success rates are utilized as before in Chapter 7 and Chapter 8. The action models are run five times. The goal models monitor these executions. Both outcomes are compared with the judgements of the experimenter

on whether the skill was successful or not.

In addition, the users were asked to complete an exit survey. The users were asked a multiple choice question with an optional comment section: "Would you have preferred more granular feedback such as *close success*, *complete success*, *near miss* and *complete fail*? If so, what are your suggestions for what kind of granular feedback would have been useful in your experience?". The choices were "Prefer more granular feedback", "Prefer binary feedback" and "No strict preference". The users were also presented with the multiple-choice questions to compare the demonstrations and exploration phases based on "Preference", "Usefulness", "Ease of Interaction", "Wanting to spend more time" and "Wanting to spend less time" with the options of "Demonstration", "Exploration", "Both" and "Neither". In addition, the following open ended questions were asked:

- Would you briefly describe your teaching strategy? Did seeing the robot execute the learned action change anything about the strategy you started with?
- How exactly did you decide on the success/fail when the robot attempted to execute a skill?
- Do you have any other suggestions or comments about your teaching experience with Curi?

9.3.3 Perception System Issues

The perception system was first developed for Experiment III (Chapter 7). Then it was updated to have better performance for finding the right object for the evaluation described in Chapter 8.

The driver for the RGBD camera does not allow to set the exposure and white balance manually. This sometimes results in color shifts, overexposure (*e.g.* bright colors becoming white) and underexposure (resulting in false color and noise). Since object selection is

dependent on color, this adversely effects the performance. A manual step on top of the automatic object segmentation was done in Experiment III (Chapter 7). The updated system was able to handle the challenges posed by these issues for the evaluation in Chapter 8.

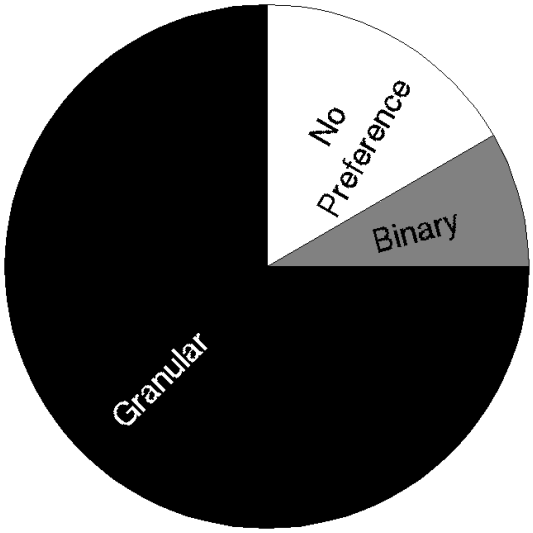
For this experiment, there were a few additional issues with the perception system that was noticed after the fact. These issues affected the goal model performance. This experiment uses the same perception system was used with the same parameters as Sec. 8.3.1. However, for this experiment, the robot's position was different which led the robot to cast a shadow on the object. This further complicated the sensor related issues. In addition, macaroni was switched to penne for the pour skill. Penne spreads more than macaroni due to its shape, making the surface of the pasta-filled bowl flat. This made is harder to detect a difference between the empty and pasta-filled bowl. This did not cause a problem during the debugging of the implementation but it came up during the experiments. As a result, the performance of the perception is worse for this study then the other instances. It is important to note that the perception system was never fine-tuned for a particular skill-object combination and some objects changed between studies.

9.4 Survey Results

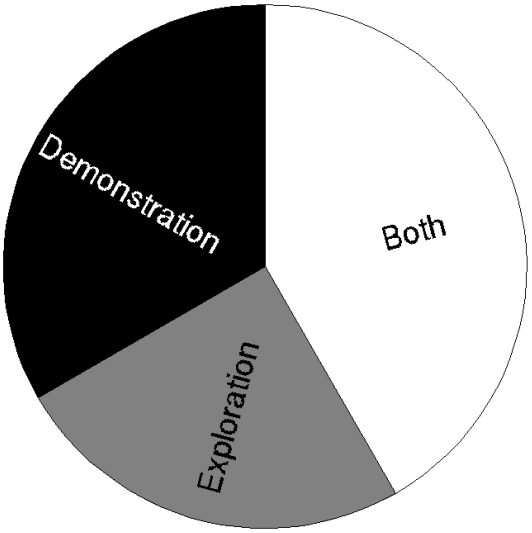
9.4.1 Multiple Choice Responses

The survey results are compiled in Fig. 47. The following results stood out from the participant responses:

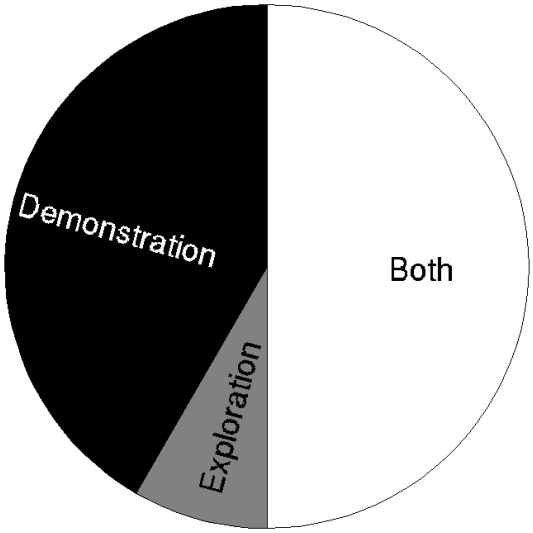
Participants prefer more granular feedback: The majority of the participants, 9 out of 12, responded in favor of more granular feedback. Only 1 user preferred binary feedback as used in the experiment. 2 users did not have any preference. The results are shown in Fig. 47(a). 5 participants who preferred granular feedback agreed with the proposed responses; *close success*, *complete success*, *near miss*, and *complete fail*. 2 participants explicitly mentioned that there should not be too many and 1 participants made a comment about the difficulty of quantifying the granular responses.



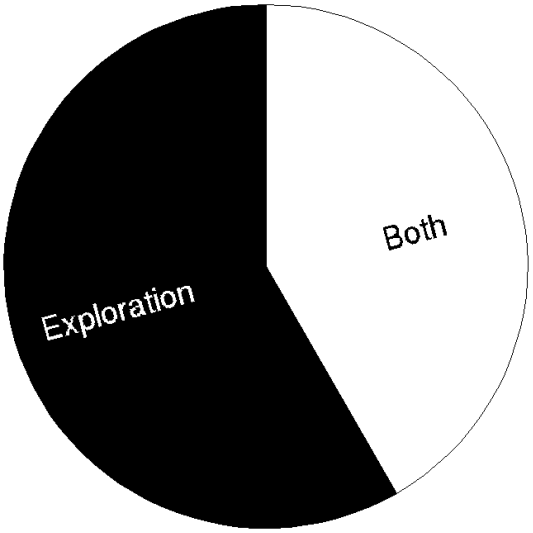
(a) Granularity



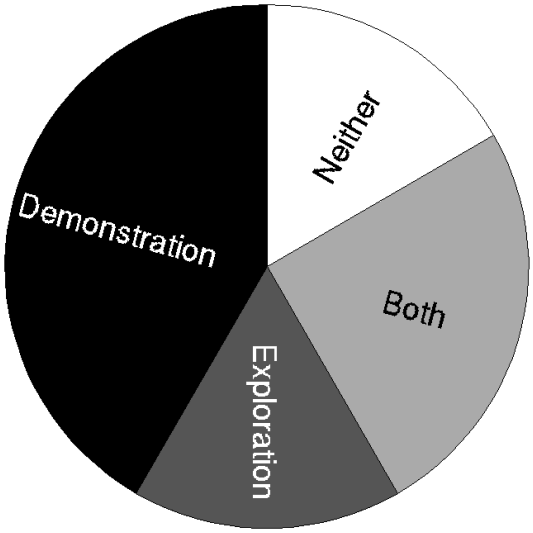
(b) Preference



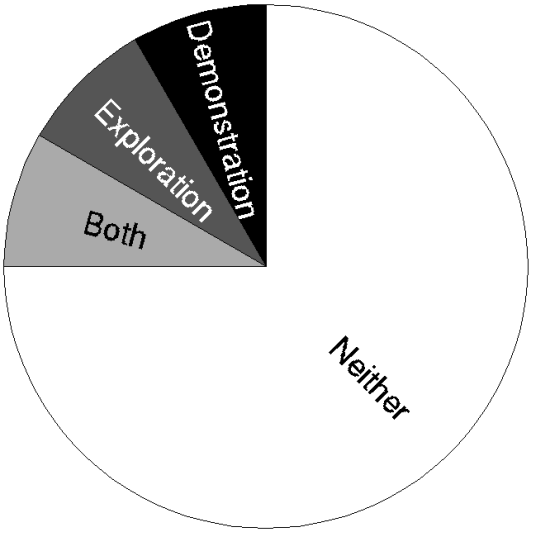
(c) Usefulness



(d) Ease



(e) Wanting to spend more time



(f) Wanting to spend less time

Figure 47: The survey results for the multiple choice questions

Participants were positive towards both teaching phases: There was no significant difference between the participants preferences as shown in Fig. 47(b). Both phases were found to be useful with a slight edge for demonstrations (see Fig. 47(c)). No participant gave any negative response to the questions of *Preference* and *Usefulness*. However, due to low number of participants, no conclusions can be drawn here.

More participants thought the exploration phase was easier: This is shown by the fact that 7 users thought exploration was easier. The remaining 5 thought both were equally easy. No user thought demonstration was easier than exploration and no user found both phases difficult as shown in Fig. 47(d).

Overall participants wanted to spend more time teaching: The survey responses presented in Fig. 47(e) and Fig. 47(f) show that only three participants did not want to spend more time in any of the phases. The data does not provide evidence to conclude a significant difference between the two phases that the participants wanted to spend more time on.

9.4.2 Open Ended Responses

The survey asked two open ended questions to the participants regarding their teaching strategy and their decisions about robot's execution success. 7 users explicitly stated that they changed their teaching strategy after seeing the robot execute the action. Only 1 user explicitly stated he tried to keep his strategy constant.

The participants responded by stating that both the open the box and close the box skills have binary outcomes so it was straightforward to decide on success for those skills. However, there were a few cases where this did not hold for the open the box skill. One such example is shown in Fig. 46. The user told the robot that this was successful. However, another user in the same situation told the robot that it failed. 2 users gave feedback based on robot's execution being similar to demonstrations rather than goal success for their demonstrated skills. These users forgot to provide intermediate keyframes and as a result robot

failed. However, users gave positive feedback, thinking that the failure was on their part and not on the robot. One of these user stated "I based success/fail on whether the robot executed the instructions in the same way that I executed the instructions with her.".

For the pour skill, participants were told to pour most of the pasta into the bowl. This was reflected in their responses as being strict. Some responses include "90% of the pasta in the bowl", "At most three pieces out of the bowl", and "all the pasta in the bowl". 1 user only thought to pour half of the pasta and was okay with this. This user responded by saying that she did not care about sloppiness.

Two users stated that they also considered the robot not damaging anything in its executions as being part of success.

In the general comments section, one user realized that giving a varied set of demonstrations during the demonstration phase, affects how the robot samples in the exploration phase.

9.5 Action and Goal Learning Performance

9.5.1 Skill Execution

This section presents the execution performance of the action models after the experiment. This section follows a methodology similar to Sec. 7.4. To compare the effects of the interaction, two action models for each skill is tested; (1) learned only using *demonstration* data (5 data points) and (2) learned using *both* demonstration and sampled data that was marked as success (5-10 data points, depending on user feedback). The skills are executed five times using these models for each of the three skills and each of the 12 participants, for a total of 360 executions. The results are presented in Table 11. This section will look at the cases where there is at least 40% success rate change between the two action models.

Close the box execution success: The average success rate for the close the box skill is the same, 66.7% for both of the action models. The results show that participant 1's

Table 11: Execution success of action models learned with only *demonstration* data and learned with *both* demonstration and sampled data.

	Close the Box		Pour		Open the Box	
	Demo	Demo+Sample	Demo	Demo+Sample	Demo	Demo+Sample
1	60	100	100	100	40	100
2	100	100	100	80	0	0
3	0	0	100	100	0	60
4	100	100	100	100	0	0
5	20	20	100	100	80	80
6	60	60	80	100	0	0
7	0	0	100	100	0	0
8	80	60	100	100	0	0
9	100	60	100	100	0	0
10	100	100	80	100	0	0
11	80	100	60	100	0	0
12	100	100	100	100	60	100
Avg	66.7	66.7	93.3	98.3	15	28.3

skill success increased from 60% to 100%. This participant's demonstrations had the end-effector forcefully contact the box. This resulted in a model with low success rate; sometimes the robot would push the box away while trying to close it. Some of the sampled executions were able to push the model towards a more successful region by eliminating the forceful contact. On the other hand, the execution success of the participant 9's action model dropped from 100% to 60%. Three of the sampled executions were borderline failures which skewed to model towards this direction. The borderline cases led to a model that would sometimes fail. There were six participants who had 100% and two participants who had 0% success rate after the sampling phase.

Pour execution success: The average success rate for the pour skill increases from 93.3% to 98.3% when the sampled executions are included. The success rate of participant 11's action model increased from 60% to 100%. The action model learned just with the demonstration data would start rotating the end-effector early which sometimes resulted in spilling half of the contents. The sampled trajectories fixed this. There were 11 participants who had 100% success rate after the sampling phase.

Open the box execution success: For the open the box skill the success rate increased from 15% to 28.3%. These numbers are not as high as the previous s kills but the increase in the success is significant. There are three participants who had increased success between their action models: Participant 1's success rate increased from 40% to 100%, participant 3's from 0% to 60% and participant 12's from 60% to 100%. The main reason for these increase in success rates is that the effective stiffness of the arm is different between kinesthetic teaching and execution. This is important for the open the box skill. As a result, the successful executed action without the user touching the robot provides much better data. There were two participants who had 100% and eight participants who had 0% success rate after the sampling phase.

Overall, the effects of sampling with user feedback was either neutral or helpful. Combined with the fact that participants found this step easy (Fig. 47(d)), feedback is concluded to be a valuable interactive addition to the method. The success rate increased the most for the open the box skill.

There were only two participants (1 and 12) who had 100% success rates for all three skills.

9.5.2 Skill Monitoring

This section presents the monitoring performance of the goal models. Three goal models for each skill are tested: (1) learned only using *demonstration* data (5 data points), (2) learned using both demonstration and *execution* data (5-8 data points based on user feedback) and (3) learned using demonstration, execution and sampled data (5-13 data points based on user feedback). These goal models are tested on all the skill executions from Sec. 9.5.1 resulting in 10 test points for each model resulting in a total of 1080 (360×3) tests. The results are presented in Table 12.

The goal models success rate for the close the box skill were 67.5%, 80.83% and 87.5%. There was a steady increase which is expected since more data should produce a better

Table 12: Monitoring success of goal models learned with only *demonstration* data, with demonstration and *execution* data and with *all* of the demonstrations, execution and sampling data

	Close the Box			Pour			Open the Box		
	D	D+E	All	D	D+E	All	D	D+E	All
1	90	90	90	90	90	90	50	50	50
2	100	100	100	60	60	60	100	100	100
3	100	100	100	80	80	90	90	90	90
4	100	100	100	50	50	50	100	60	60
5	80	20	60	80	100	100	70	90	90
6	60	60	60	90	90	90	100	100	100
7	100	100	100	90	90	90	100	100	100
8	30	70	80	60	60	80	100	100	100
9	20	40	60	70	100	100	100	100	100
10	30	100	100	70	70	90	100	100	100
11	10	90	100	80	80	80	100	100	100
12	90	100	100	40	80	80	60	60	70
Avg	67.5	80.83	87.5	71.7	79.2	83.3	89.2	87.5	88.3

model. There was a significant increase for participant 11, from 10% to 100% after learning with all the data. Conversely, for participant 5, the success dropped from 80% to 20% before increasing to 60%. The goal model recognized failures as success. This was not an expected results since correctly labelled data should always have a positive contribution to the goal model. The most likely culprit for bad data is the perception system in this case as the participant's feedback was adequate. At the end, seven users had 100% success rate for their goal models.

For the pour skill, the rates increased steadily; 71.7%, 79.2% and 83.3%. There were no significant changes between the results. There were three participants who had at least 90% success rate. One participant had 50% success rate. However the results are a bit misleading, as explained in Sec. 9.3.3, since the perception system was not reliable for this skill. Thus, the results should be taken with a grain of salt.

The open the box goal model success rates were steady at 89.2%, 87.5% and 88.3%. Nine users had at least 90% success rate and one user had 50% success rate. The goal model for participant 4 shows an anomaly, the success rate goes down from 100% to 60%.

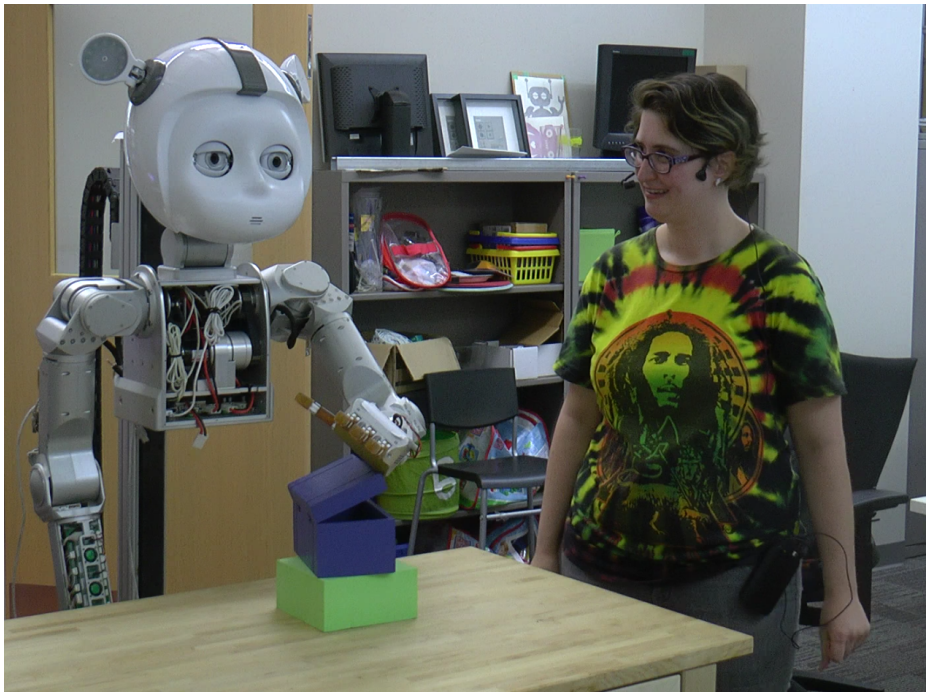


Figure 48: The robot at the end of an execution for open the box skill. The robot was not able to fully open the box. The participant told the robot that it succeeded.

The main reason is that this participant provided feedback which made it difficult for the goal model to discern between a closed and an open box. The box was only opened slightly, which was impossible for the perception system to handle. An example of this is illustrated in Fig. 48.

9.5.3 Comparison with Experiment III

Table 13 presents the results for both of the models with the five executions performed using the action and goal models learned with all the available data. The *Exec* columns are the same as the *Demo+Sampled* columns of Table 11. The data in this table is presented the same way as Table 9 in Chapter 7.

The results show that the action success rates are higher for both the close the box (57.5% vs 66.7%) and the pour (75% vs 98.3%) skills as compared to Experiment III described in Chapter 7. There are three reasons for this increase. First, there was no object orientation difference between demonstrations in this experiment as illustrated in

Table 13: Skill execution and monitoring results. The results are obtained by executing the action models learned from all the data five times. Similarly, monitoring on these executions are done using the goal models learned from all available data.

	Close the Box			Pour			Open the Box		
	Exec	Monitoring		Exec	Monitoring		Exec	Monitoring	
		True	False		True	False		True	False
		P : N	P : N		P : N	P : N		P : N	P : N
1	100%	4 : 0	0 : 1	100%	5 : 0	0 : 0	100%	2 : 0	0 : 3
2	100%	5 : 0	0 : 0	80%	3 : 0	1 : 1	0%	0 : 5	0 : 0
3	0%	0 : 5	0 : 0	100%	4 : 0	0 : 1	60%	2 : 2	0 : 1
4	100%	5 : 0	0 : 0	100%	1 : 0	0 : 4	0%	0 : 2	3 : 0
5	20%	1 : 1	3 : 0	100%	5 : 0	0 : 0	80%	4 : 1	0 : 0
6	60%	1 : 2	0 : 2	100%	5 : 0	0 : 0	0%	0 : 5	0 : 0
7	0%	0 : 5	0 : 0	100%	5 : 0	0 : 0	0%	0 : 5	0 : 0
8	60%	2 : 2	0 : 1	100%	5 : 0	0 : 0	0%	0 : 5	0 : 0
9	60%	1 : 0	2 : 2	100%	5 : 0	0 : 0	0%	0 : 5	0 : 0
10	100%	5 : 0	0 : 0	100%	5 : 0	0 : 0	0%	0 : 5	0 : 0
11	100%	5 : 0	0 : 0	100%	5 : 0	0 : 0	0%	0 : 5	0 : 0
12	100%	5 : 0	0 : 0	100%	4 : 0	0 : 1	100%	3 : 0	0 : 2
Overall	66.7%	34 : 15	5 : 6	98.3%	52 : 0	1 : 7	28.3%	11 : 40	3 : 6
		81.7%			86.7%			85%	

Fig. 34. This resulted in more consistent demonstrations. Second, the users were able to see the action models through robot executions. Some users modified their demonstrations to teach a better model.The third reason is the modification of the action execution to favor HMM state paths that are closer in length to the average number keyframes observed during demonstrations.

The goal model success rates are between 80% and 90% for this experiment as compared to 90% for Experiment III and 92% for the evaluation in Chapter 8. This slight decrease in perfomance is due to the bad performance of the perception system. Overall, Experiment IV replicated the results of Experiment III. This shows again that action and goal models can be simultaneously learned from the same set of non-expert teacher demonstrations.

9.6 Self-Improvement after User Interactions

Chapter 8 introduced a method to increase the execution success of the learned action models based on successful goal model performance. In this experiment, the average execution success was 64.4% after learning with the teachers. This suggests that the action model success should be improved. The average goal monitoring success (86.4%) suggests that this is possible. The sampling phase with participants (only 5 data points) was not long enough to fix the cases where they were not able to teach a successful action model. Another self-improvement phase using the participant data as seed is needed to increase the execution results. In this section, the robot's ability to self-improve using models learned from non-expert teacher data will be evaluated.

9.6.1 Data

The requirement for self-improvement is to have a successful goal model and it makes the most sense to improve severely failing action models. As a result, the cases where the action model had at most 60% execution and the goal model had at least 90% monitoring success were chosen for self-improvement. This resulted in five cases; two close the box and three open the box. All of the participants' execution successes for the pour skill was over 90%, so this skill was not included in this evaluation.

Each self-improvement case had 9 episodes with 5 roll-outs each resulting in 225 runs ($5 \times 9 \times 5$). In addition, action models learned after each episode were executed to get the success rates of the models, resulting in another 225 runs. Users demonstrations were *forgotten* after getting 10 successful samples as explained in Chapter 8. Note that all the success labels come from the goal models for the purposes of self-improvement.

9.6.2 Metrics

In addition to the execution success, the distances between the initial execution path and the execution path obtained from action models after each episode are calculated to measure

the change in the models during self-improvement. The initial path is obtained from the action model learned using the data (demonstrated and sampled during the interaction) provided by the participant. Recall that the path to be executed is generated by finding the most likely state path between the prior and terminal states of the action HMM and utilizing the emission means at each state (see the Alg. 1 and explanations in Sec. 6.4.1 for details). The distance between two paths is calculated by summing the distance between corresponding emission means.

Let ϕ represent the HMM state path calculated from an action model and T represent the corresponding emission distribution mean sequence. For example, let $\phi = \{s_1, s_5, s_3\}$ where s_j is the j^{th} state which leads to the corresponding emission mean sequence $T = \{\mu_1, \mu_5, \mu_3\}$. For the purposes of this section, it is assumed that all the state paths (*i.e.* ϕ's) obtained from the action models of a participant for the same skill have the same number of states. Then distance metric between two paths is defined as follows:

$$D(\phi_i, \phi_j) = \sum_{r=1}^{k} d(T_i^r, T_j^r) \tag{2}$$

In, Eq. 2, k represents the number of states along the paths. Note that $k \leq n$ where n is the smallest number of states in the corresponding HMMs. Furthermore, subscripts denote the HMM membership, superscripts denote the corresponding element of a sequence (*e.g.* $T^2 = \mu_5$ from the previous example) ,and $d(\cdot,\cdot)$ represent the distance between two emission means which is given in Eq. 5.

This is not a general distance between two HMMs or two paths and only works if the paths have the same length. A Dynamic Time Warping (DTW) approach could be used to handle paths with different length. It was not necessary for the purposes of this section. A more general metric would be the Kullback-Leibler Divergence between HMMs which includes the effects of the covariances and the HMM structure. However, this is a hard problem where only approximations exists.

The metric in Eq. 2 involves distances between two rigid body poses which is denoted

with $d(\cdot,\cdot)$. Recall that throughout this thesis, a 3-dimensional vector was used to represent the translational component and a unit quaternion (4-dimensions) to represent the rotational component of a rigid body pose, resulting in a 7-dimensional vector (see Sec. 6.2.2). Since a unit quaternion and its antipodal represent the same rotation, the Euclidean distance in this 7D space is not appropriate; calculating the distance this way would yield different results depending on how the unit quaternion is oriented. Unfortunately, there is no intrinsic metric in the space of rigid body transformations (*i.e.* the $SE(3)$ manifold) to calculate distances. Instead, the distances between the translational and rotational parts are computed separately between the two poses. The translational part is the Euclidean distance in $\mathbb{R}^3$. The rotational part is the minimum angular rotation (in radians) between two rotations. This is the intrinsic metric of the space of rigid body rotations ($SO(3)$).

Let $\mu = [\mu_x; \mu_q]$, where the vector μ is the concatenation of vectors μ_x and μ_q. μ represents a 7D emission mean of an action model, μ_x represents its 3D translational component and μ_q represents its 4D rotational component as a unit quaternion. The translational distance is calculated as follows:

$$d_x(\mu_x^i, \mu_x^j) = \sqrt{(\mu_x^i - \mu_x^j)^T(\mu_x^i - \mu_x^j)} \tag{3}$$

Thee rotation between two unit quaternions can also be represented by a third unit quaternion. Let q_1 and q_2 represent two unit quaternions. Then the quaternion that represents the minimum rotation between these two is given by $q_{diff} = q_1^{-1} q_2$ where the inverse operation is the quaternion inverse and multiplication operation is the quaternion multiplication. Then, the scalar part of q_{diff} equals the cosine of the half of the angle between the two ($q_{diff}.w = cos(\theta/2)$, where $.w$ represents the scalar). The quaternion distance is calculated as follows:

$$d_q(\mu_q^i, \mu_q^j) = 2acos(((\mu_q^i)^{-1}\mu_q^j).w) \tag{4}$$

Then, the distance between two means (*i.e.* rigid body poses) is calculated as:

$$d(\mu^i, \mu^j) = d_x(\mu_x^i, \mu_x^j) + \beta d_q(\mu_q^i, \mu_q^j) \quad (5)$$

The parameter β sets the relative importance of the rotation part with respect to the translation part. There is no *intrinsic* way to select this parameter so it is selected based on the application. For the purposes of this thesis, it is selected as $\beta = 1$.

9.6.3 Case Studies

9.6.3.1 Participant 3 - Close the Box

The top of the Fig. 49 shows the execution success rates of the action models learned from participant 3 and then self-improved. After both demonstrations and sampling, the action model had 0% execution success and the goal model had 100% monitoring success. Self-improvement was able to increase the execution success to 80% with one episode. By the end of the 2^{nd} episode, the action model had 100% success. The bottom of the Fig. 49 shows the distances between the initial execution path and the execution paths obtained from the action models at each episode. A total of $5cm$ difference for 5 states was enough to get the model from 0% success to 80% success which suggests that the models were close to begin with even though the execution success was 0%. Note that the difference jumps after the participant data is forgotten.

9.6.3.2 Participant 7 - Close the Box

The Fig. 50 shows the results for the close the box action models from self-improvement, seeded with participant 7. The self-improvement started with 0% execution success and 100% monitoring success. In this case, the self-improvement was not able to influence the action model until after the 5^{th} episode. The distance to the initial path graph almost mimics the success rate graph. A large change was needed to improve the model. This large change came from both the adaptive sampling (the method looked farther and farther out since it was failing most of the time) and from the information that was already in the action HMM. The samples that were able to fix this model came from a different state path

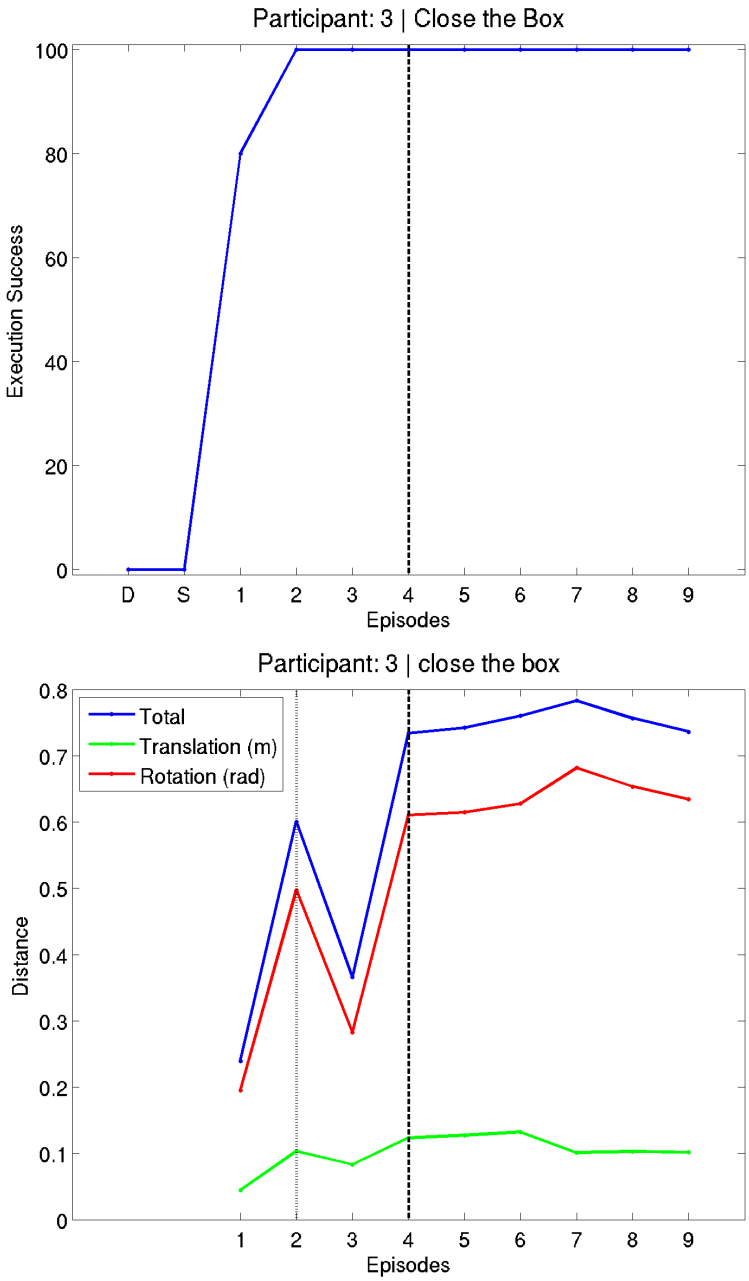


Figure 49: The close the box (top) execution success rates and (bottom) executed path distances against the initial path for participant 3. The horizontal axis represents the action models at different steps. **D** represents the action model learned only with participant demonstrations. **S** represents the one learned by including the sampled data with feedback. The numbers represent the episode of the self-improvement. The dashed line represents the point of *forgetting* teacher demonstrations and the dotted line represents the point of getting 100% execution success.

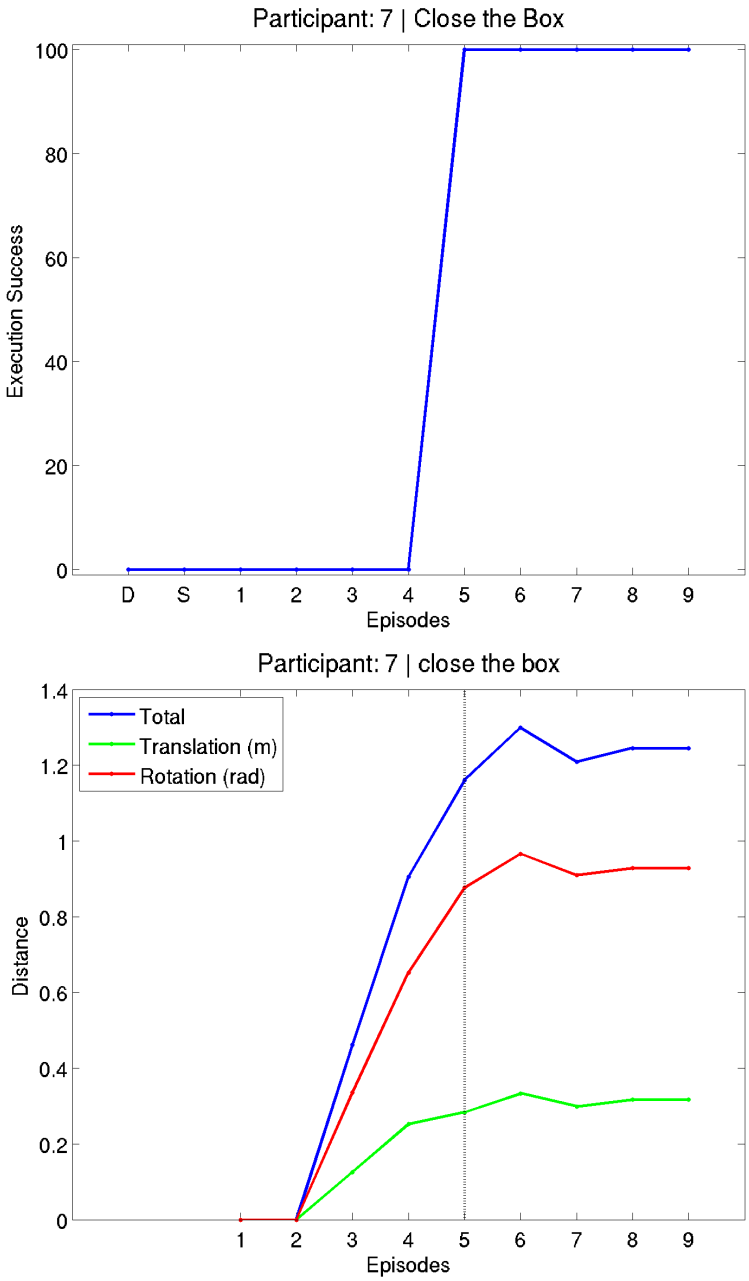


Figure 50: The close the box (top) execution success rates and (bottom) executed path distances against the initial path for participant 7. The horizontal axis represents the action models at different steps. **D** represents the action model learned only with participant demonstrations. **S** represents the one learned by including the sampled data with feedback. The numbers represent the episode of the self-improvement. The dotted line represents the point of getting 100% execution success. There were not enough number of successful samples to *forget* the teacher demonstrations.

than the most likely one. This shows that the addition of state path sampling was useful in this case. Since the successful paths from a different topological path obtained with big sampling steps, the differences between the paths are large (more than 30*cm* and 0.9*rad*). The initial path was longer than the final path (5 vs 4) of the last two action models. As a result, the last frame was omitted from all the paths. Also note that there were not enough number of successful samples to *forget* participant demonstrations.

9.6.3.3 Participant 11 - Open the Box

The Fig. 51 shows the results for the open the box action models from self-improvement, seeded with participant 11. The self-improvement started with 0% execution success and 100% monitoring success. The self-improvement method struggled was able to slightly increase the success but was not able to find a good model until episode 6. However,the required changes were not too high, around 5*cm* and 0.15*rad* for 3 states. Note that the difference jumps after the participant data is forgotten. However, the change continued to increase after this point as well.

9.6.3.4 Participant 3 - Open the Box

The Fig. 52 shows the results for the open the box action models from self-improvement, seeded with participant 3. The self-improvement started with 60% execution success and 90% monitoring success. There was 1 successful sampled during episode 1 that changed the model but it was not enough to increase the success rate above 60%. The self-improvement method was not able to change the model until after episode 4. After episode 6, the success rate was 100%. The differences suggest that the models were already relatively close. Forgetting the user data did not have a significant impact.

9.6.3.5 Participant 7 - Open the Box

The Fig. 53 shows the results for the open the box action models from self-improvement, seeded with participant 7. The self-improvement started with 0% execution success and

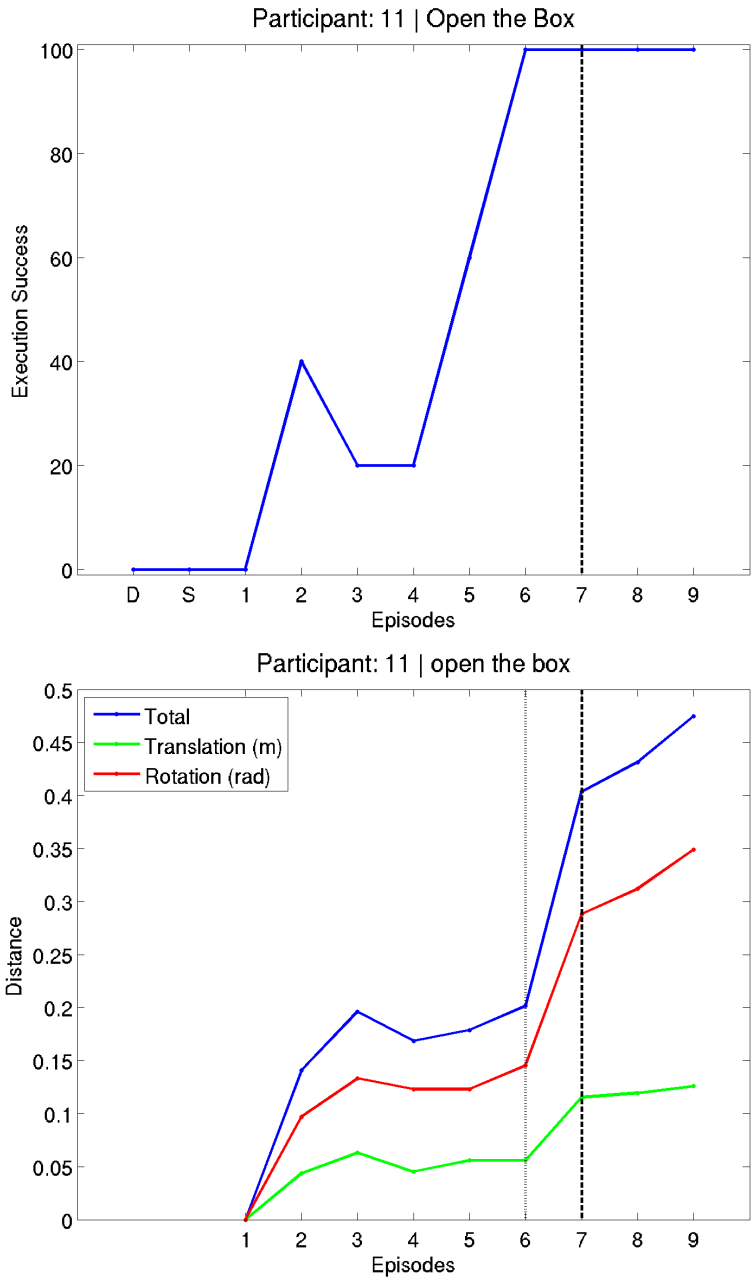


Figure 51: The open the box (top) execution success rates and (bottom) executed path distances against the initial path for participant 11. The horizontal axis represents the action models at different steps. **D** represents the action model learned only with participant demonstrations. **S** represents the one learned by including the sampled data with feedback. The numbers represent the episode of the self-improvement. The dashed line represents the point of *forgetting* teacher demonstrations and the dotted line represents the point of getting 100% execution success.

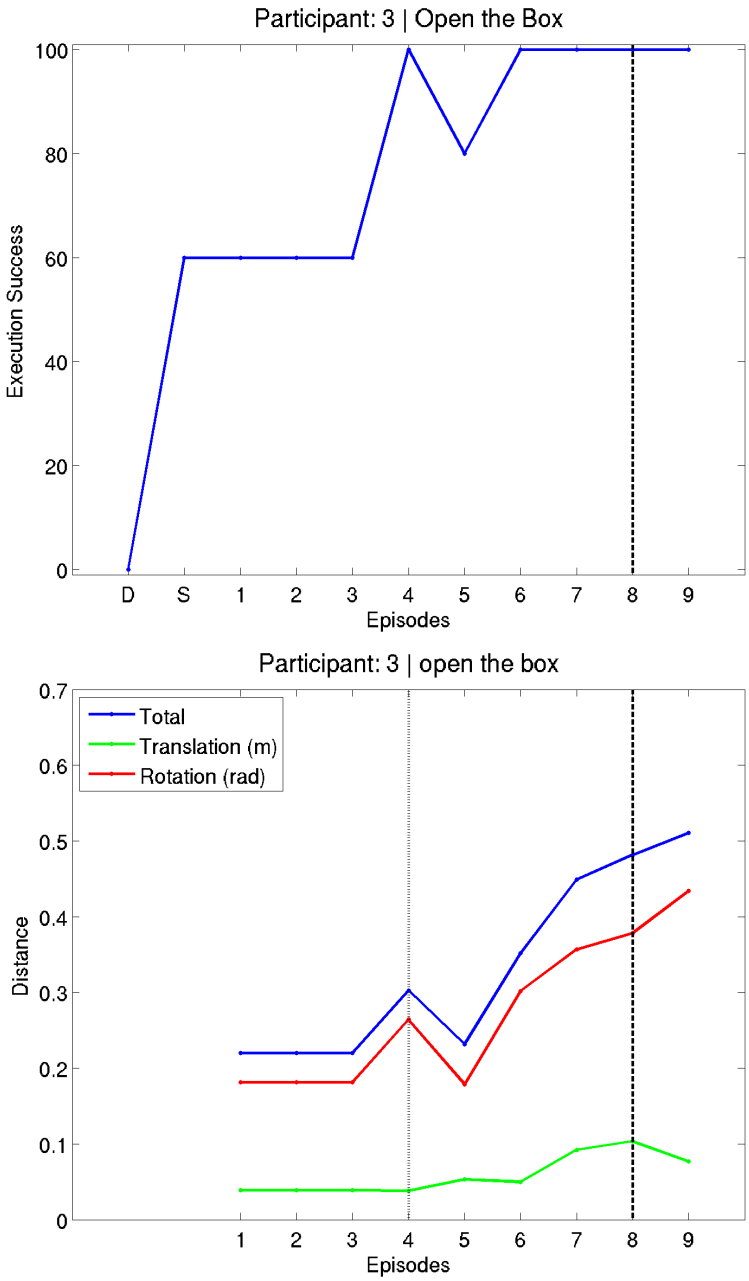


Figure 52: The open the box (top) execution success rates and (bottom) executed path distances against the initial path for participant 3. The horizontal axis represents the action models at different steps. **D** represents the action model learned only with participant demonstrations. **S** represents the one learned by including the sampled data with feedback. The numbers represent the episode of the self-improvement. The dashed line represents the point of *forgetting* teacher demonstrations and the dotted line represents the point of getting 100% execution success.

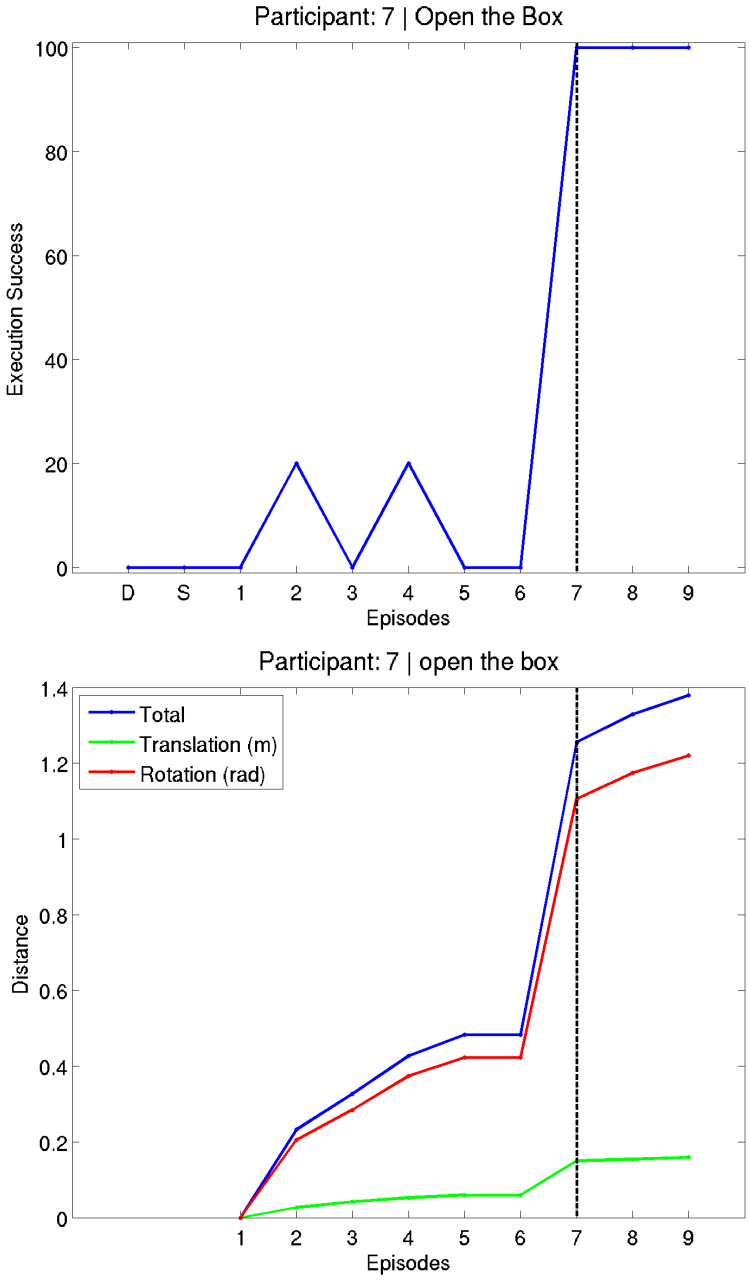


Figure 53: The open the box (top) execution success rates and (bottom) executed path distances against the initial path for participant 7. The horizontal axis represents the action models at different steps. **D** represents the action model learned only with participant demonstrations. **S** represents the one learned by including the sampled data with feedback. The numbers represent the episode of the self-improvement. The dashed line represents both the point of *forgetting* teacher demonstrations the point of getting 100% execution success since the two coincide in this case.

100% monitoring success. The self-improvement method was not able to fix the action model until the participant demonstrations were forgotten. The distance graphs shows that steady progress was made but it was necessary to forget the user demonstrations. This graph also suggests that a relatively large jump was required to fix the model. However, further analysis shows that most of the change came from the initial and final states did not have a significant impact on the success which were away from the object. The middle 2 states, among a total of 4 states, mattered more. The difference between the middle 2 states of the initial and the final model was $4.25cm$ and $0.25rad$.

9.7 Summary

This chapter introduced *Interactive Goal based Learning and Exploration - iGoaL-E* framework which added interactive executions, monitoring and feedback to the framework developed before. This was tested with non-expert teachers.

This chapter evaluated the action and goal models learned from non-expert teacher demonstrations and feedback. The execution success of the action models learned with all the available data was 64.4% averaged accross all the users and skills. The average monitoring success of the goal models was 86.4%. The results of Experiment III (Chapter 7) were also replicated. It was shown that the feedback on direct and sampled executions are helpful to get additional data from user and increase the model performance. The results of Sec. 9.4 suggests that this feedback option was also well received by the users However, it was also seen that, on rare occasions, feedback can be detrimental

This chapter also evaluated the self-improvement algorithm described in Chapter 8 with non-expert data. The additions to the model, specifically sampling from all the state paths, helped fix one severely failing case, which shows the efficacy of the method and the fact that even really bad demonstrations can have important information. As long as the action models are not too far away or the demonstrations have enough information and the goal models are successful, the GoaL-E approach can be used to improve the action model.

Overall, the results show that; (1) action and goal models can be learned from non-expert teacher data (replicated Experiment III), (2) feedback can increase execution and monitoring performance, and (3) self-improvement is possible with non-expert data. In addition, the participants were positive towards both demonstrations and providing feedback and reported that seeing the robot executions helped their demonstrations.

CHAPTER X

CONCLUSION AND FUTURE WORK

The broad vision of the this thesis is to enable everyday people to teach skills to their robots through learning from demonstration. This thesis has taken a holistic view on the entire LfD interaction and has developed an end-to-end system. The development has included the design of novel interactions and algorithms that enhance LfD. An HRI perspective was taken to address the challenges that were set out in Chapter 1. The work in this thesis started by looking at how non-experts interact with robots in an LfD setting through a series of user studies. The purpose of these experiments was to find out what the people are good at demonstrating and how is the best way to get data from them. Based on the observations, keyframes were introduced to help with demonstrations and a framework to learn from multiple types of demonstrations were developed. Further observations led to the idea of goal learning. The learned goals were then used to improve the actions necessary to execute the learned skills. Finally, an end-to-end interactive LfD framework was developed based on all the results. Overall, more than 80 people participated in four experiments to test the methods developed in this thesis. This thesis led to the following contributions that address the challenges mentioned in Chapter 1.

10.1 Summary of Contributions

10.1.1 Keyframe and Hybrid Demonstrations

This thesis introduced keyframes to the field of Learning from Demonstration. Keyframes are robust to the noisy, inconsistent and unintended demonstrations of non-expert teachers. However, they cannot communicate timing/dynamics information like trajectory demonstrations. To leverage the best of both worlds, this thesis introduced hybrid demonstrations and developed a framework called the Keyframe based Learning from Demonstration

(KLfD) to learn from these demonstrations.

In addition, keyframes allow the teachers to highlight salient parts of a skill during demonstrations which helps with goal learning as described next.

The contributions described in this section are mainly empirical results about naïve users in general and keyframes and hybrid demonstrations in particular. These contributions are mainly related with the requirements 1 and 4 put forth in Chapter 1.

10.1.2 Learning Actions and Goal Models

The experiments on LfD with everyday people showed that they concentrate on achieving the goal of the skill more so than how to exactly achieve it. The developed framework and algorithms, called *GOal and Action Learning from Demonstration - GoaLfD*, leverages this goal oriented nature of teachers and learns both action and goal models simultaneously from the same set of demonstrations. This thesis showed that the goal models have high monitoring success rate even when the learned action models do not perform as well. The contributions are mainly algorithmic supported by an empricial evaluation (with 12 users) and they are mainly related with the requirements 2, 3 and 4 put forth in Chapter 1.

10.1.3 Self-Improvement

The learned action models may not always be satisfactory or represent the variance of the skill. This thesis introduced a self-improvement method, called *Goal based Learning and Exploration - GoaL-E*, to remedy this. GoaL-E uses the probabilistic nature of the action models to execute the learned skills with variety and goal models to monitor this execution. Then, the action models are updated by utilizing the output of this monitoring. This is similar to existing reinforcement learning approaches but it does not require any reward function to be pre-programmed. In addition, adaptive sampling is employed to handle the exploration versus exploitation trade off. The contribution of the section is algorithmic and is mainly related with the requirement 2 put forth in Chapter 1.

10.1.4 Interactive Learning

The final LfD system incorporates all the aforementioned contributions and closes the loop with interactive components. By executing the learned skill and verbalizing the monitoring result, the robot implicitly communicates its state of learning. The teacher can then tailor his/her demonstrations accordingly. In addition, the teacher is able to provide verbal feedback on these communicative executions and during self-improvement to provide data with little effort. The resulting framework is called the Interactive GoaL-E or *iGoaL-E*. An user study showed that feedback is useful in learning action and goal models and non-expert data can be used to seed self-improvement. Since the final version of the framework consolidates all of the previous results, it is related with all the requirements put forth in Chapter 1.

10.2 Open Questions and Future Work

Learning from demonstration is an exciting paradigm that will enable robot end-users to program and customize their robots. This thesis is among the first step in the direction of evaluating LfD methods in the loop with non-expert teachers. There are a few open questions and interesting future work as a result of this thesis.

10.2.1 Using Learned Models in More Challenging Environments

The main aim of this thesis is to enhance LfD for non-expert teachers. This requires robots to be able to learn and function in various environments with a wide variety of teachers. The experiments in this thesis were done in laboratory settings with people mainly from the university campus community. In-situ experiments with a wider range of participants are the logical next steps in further understanding human behavior and the resulting data in LfD settings. The results will guide algorithm development and interaction design.

These experiments will be performed in more realistic scenarios which will involve multiple objects, clutter, and other people. New interaction strategies will be required to

learn the object(s) of interest and/or relevant reference frames during teaching. Moreover, planning/optimization methods will need to be integrated to handle clutter and improve generalization. An initial attempt at integrating planning is presented in [46]. However, to have a satisfactory answer to this problem requires a new PhD thesis. Furthermore, robots will need social awareness to let the other people around know what they are doing.

Tighter coupling of action execution and monitoring will be crucial for the robots in realistic settings. When the robot needs to function, it needs to know if an action is applicable and if the execution resulted in success. Goal models can be used to determine if an action is applicable (*e.g.* by looking at the prior states). If the robot fails, it can try to retry if the action is still applicable, try another action or ask for help. These will all be required since robots will fail and failure recovery is important.

10.2.2 Batch versus Interactive Teaching

There is potential for interactive learning being better than batch learning since the teachers will be able to tailor their demonstrations based on what they see. Some would argue this is not always the case since it breaks the common *i.i.d.* assumption in learning since the user demonstrations will not be independent. A systematic study of batch versus interactive learning is needed in the context of robotic LfD. This will show whether interactive learning is worth the extra effort in interaction and algorithm design. The author argues from his personal experience that a robot that at least executes its learned actions during teaching is more engaging then a robot that just collects batch data, so the benefits might not be limited to learning efficiency. The initial results will guide the development of new incremental learning algorithms that appropriately model the human data.

The specifics of the interaction will matter as well. Achieving the level of interaction in human-human teaching scenarios for human-robot teaching is a herculean task. However, there are several directions that are promising. The robot can go further than just executing its action to communicate its learning state. Transparency mechanisms let the user better

communicate its internal model. An example would be to hesitate while executing the skill to show low confidence on the learned model. In Experiment IV, the participants showed interest in granular feedback. A question is how to quantify granular feedback and how to include in an incremental update of the models. An interactive teaching paradigm affords other interesting opportunities. In Experiment I, additional keyframe interactions, such as keyframe iterations, were explored. The impact of similar interactive additions to LfD need to be explored and compared with respect to batch learning.

10.2.3 Goal Learning with Hybrid Demonstrations

The work in this thesis opted to use keyframes for goal learning to scope the problem down. However, Experiment II and AAAI LfD challenge 2011 showed that people are good at utilizing hybrid demonstrations. Another thread of future work is to explore goal learning with hybrid demonstrations.

Using the high dimensional object data trajectory would be prohibitive. As a result, first step would be to record only the starting and ending points of a trajectory segment as object data, while recording everything in between for the motor data. Other options would include, extracting keyframes from the both the motor data and object data streams together, instead of just the motor stream.

This step would fully integrate hybrid demonstrations, KLfD and iGoaL-E.

10.2.4 Self-Improvement

The self-improvement method developed in this thesis is a first step in this direction. Even with adaptive sampling, it is too close to rejection sampling. A deeper investigation is required to develop more efficient methods. In addition, links to existing work such as (Inverse) Reinforcement Learning should be made stronger. Another issue is to forgetting of user data. A more structured approach, *e.g.* based on information metrics, to decide on whether to remove user data or not should be developed.

An idea to make it more efficient is to utilize failed data. Currently, the decision of

the goal model at the end of the execution is utilized for self-improvement. However, monitoring can be done throughout the execution. The successful parts of a failed execution can be used to make local updates to the action model. Other sampling strategies can be utilized. An example would be to systematically sample from states as a means to only update those states.

10.3 Final Remarks

To the best of our knowledge, this thesis develops the first end-to-end interactive LfD system that can self-improve while making experimental and algorithmic contributions. The subcomponents of the system has been tested with everyday people who are not robotic experts. The results suggest that the everyday people can use keyframes to teach action and goal models to the robot through kinesthetic teaching. The goal models can successfully monitor the skill executions. The system can further improve the learned skill on its own.

This thesis, has taken a Human Robot Interaction perspective on Learning from Demonstration. The most interesting lesson that the author learned early on is that everyday people never act the way that the roboticist expects them to. The assumptions and internal models are very different. Furthermore, everyday people tend anthropomorphise the robots much more than the roboticists. These are the exact reasons that the roboticists should include the intended users in the loop for developing robotic applications.

REFERENCES

[1] ABBEEL, P. and NG, A., "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the 21st Intl. Conference on Machine Learning (ICML)*, pp. 1–8, 2004.

[2] ABBEEL, P., COATES, A., and NG, A. Y., "Autonomous helicopter aerobatics through apprenticeship learning," *The Intl. Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.

[3] AKGUN, B., CAKMAK, M., JIANG, K., and THOMAZ, A. L., "Keyframe-based learning from demonstration," *Intl. Journal of Social Robotics*, vol. 4, no. 4, pp. 343–355, 2012.

[4] AKGUN, B., CAKMAK, M., WOOK YOO, J., and THOMAZ, L. A., "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," in *Intl. Conference on Human-robot interaction (HRI)*, pp. 391–398, 2012.

[5] AKGUN, B., JIANG, K., CAKMAK, M., and THOMAZ, A., "Learning tasks and skills together from a human teacher," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pp. 1868–1869, The AAAI Press, 2011.

[6] AKGUN, B., SUBRAMANIAN, K., and THOMAZ, A., "Novel interaction strategies for learning from teleoperation," in *AAAI Fall Symposia 2012, Robots Learning Interactively from Human Teachers*, 2012.

[7] AKGUN, B. and THOMAZ, A., "Learning constraints with keyframes," in *Robotics: Science and Systems: Workshop on Robot Manipulation*, 2013.

[8] AKGUN, B. and THOMAZ, A., "Self-improvement of learned action models with learned goal models," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[9] AKGUN, B. and THOMAZ, A., "Simultaneously learning actions and goals from demonstration," *Autonomous Robots, Online First*, 2015.

[10] AMOR, H. B., BERGER, E., VOGT, D., and JUN, B., "Kinesthetic bootstrapping: Teaching motor skills to humanoid robots through physical interaction," *Lecture Notes in Computer Science: Advances in Artificial Intelligence*, vol. 58, no. 3, pp. 492–499, 2009.

[11] ARGALL, B., CHERNOVA, S., VELOSO, M. M., and BROWNING, B., "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[12] ATKESON, C. G. and SCHAAL, S., "Robot learning from demonstration," in *Proc. 14th Intl. Conference on Machine Learning*, pp. 12–20, Morgan Kaufmann, 1997.

[13] BAUM, L., PETRIE, T., SOULES, G., and WEISS, N., "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The annals of mathematical statistics*, pp. 164–171, 1970.

[14] BILLARD, A., CALINON, S., DILLMANN, R., and SCHAAL, S., *Robot Programming by Demonstration*, ch. 59. Springer, Dec. 2008.

[15] BILLARD, A., CALINON, S., and GUENTER, F., "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous System*, vol. 54, no. 5, pp. 370–384, 2006.

[16] BITZER, S., HOWARD, M., and VIJAYAKUMAR, S., "Using dimensionality reduction to exploit constraints in reinforcement learning," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ Intl. Conference on*, pp. 3219–3225, 2010.

[17] CAKMAK, M., *Guided Teaching Interactions with Robots: Embodied Queries and Teaching Heuristics*. PhD thesis, Georgia Institute of Technology, 2012.

[18] CALINON, S. and BILLARD, A., "Incremental learning of gestures by imitation in a humanoid robot," in *In Proc. of the ACM/IEEE Intl. Conference on Human-Robot Interaction*, pp. 255–262, 2007.

[19] CALINON, S. and BILLARD, A., "What is the teacher's role in robot programming by demonstration? - Toward benchmarks for improved learning," *Interaction Studies. Special Issue on Psychological Benchmarks in Human-Robot Interaction*, vol. 8, no. 3, 2007.

[20] CALINON, S. and BILLARD, A., "Statistical learning by imitation of competing constraints in joint space and task space," *Advanced Robotics*, vol. 23, no. 15, pp. 2059–2076, 2009.

[21] CALINON, S., GUENTER, F., and BILLARD, A., "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, vol. 37, no. 2, pp. 286–298, 2007.

[22] CHAO, C., CAKMAK, M., and THOMAZ, A., "Towards grounding concepts for transfer in goal learning from demonstration," in *Proceedings of the Joint IEEE Intl. Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*, vol. 2, pp. 1–6, IEEE, 2011.

[23] CHERNOVA, S. and VELOSO, M., "Interactive policy learning through confidence-based autonomy," *Journal of Artificial Intelligence Research*, vol. 34, 2009.

[24] CHERNOVA, S. and THOMAZ, A. L., *Robot Learning from Human Teachers*. Morgan & Claypool Publishers, 2014.

[25] CSIBRA, G., "Teleological and referential understanding of action in infancy," *Phil. Trans. The Royal Society of London*, vol. 358, pp. 447–458, 2003.

[26] DANTAM, N., ESSA, I., and STILMAN, M., "Linguistic transfer of human assembly tasks to robots," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ Intl. Conference on*, 2012.

[27] DEISENROTH, M. P., NEUMANN, G., PETERS, J., and OTHERS, "A survey on policy search for robotics.," *Foundations and Trends in Robotics*, vol. 2, no. 1-2, pp. 1–142, 2013.

[28] DIANA, C. and THOMAZ, A. L., "The shape of simon: creative design of a humanoid robot shell," in *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pp. 283–298, ACM, 2011.

[29] EKVALL, S. and KRAGIC, D., "Robot learning from demonstration: a task-level planning approach," *Intl. Journal of Advanced Robotic Systems*, vol. 5, no. 3, pp. 223–234, 2008.

[30] FINN, C., TAN, X. Y., DUAN, Y., DARRELL, T., LEVINE, S., and ABBEEL, P., "Learning visual feature spaces for robotic manipulation with deep spatial autoencoders," *CoRR/arXiv*, vol. abs/1509.06113, 2015.

[31] FLASH, T. and HOGAN, N., "The coordination of arm movements: an experimentally confirmed mathematical model," *The journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[32] GRIBOVSKAYA, E. and BILLARD, A., "Learning nonlinear multi-variate motion dynamics for real- time position and orientation control of robotic manipulators," in *Proceedings of IEEE-RAS Intl. Conference on Humanoid Robots*, 2009.

[33] HERSCH, M., GUENTER, F., CALINON, S., and BILLARD, A., "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463–1467, 2008.

[34] HOKAYEM, P. F. and SPONG, M. W., "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, 2006.

[35] HOVLAND, G., SIKKA, P., and MCCARRAGHER, B., "Skill acquisition from human demonstration using a hidden markov model," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE Intl. Conference on*, vol. 3, pp. 2706–2711, Ieee, 1996.

[36] HOWARD, A. and PARK, C. H., "Haptically Guided Teleoperation for Learning Manipulation Tasks," June 2007.

[37] HSIAO, K. and LOZANO-PEREZ, T., "Imitation learning of whole-body grasps," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 5657–5662, IEEE, 2006.

[38] JÄKEL, R., SCHMIDT-ROHR, S. R., RÖHL, S. W., KASPER, A., XUE, Z., and DILLMANN, R., "Learning of planning models for dexterous manipulation based on human demonstrations," *Intl. Journal of Social Robotics, Special Issue on Robot Learning from Demonstration*, 2012.

[39] JENKINS, O., MATARIC, M., WEBER, S., and OTHERS, "Primitive-based movement classification for humanoid imitation," in *Proceedings, First IEEE-RAS Intl. Conference on Humanoid Robotics (Humanoids-2000)*, 2000.

[40] KHANSARI-ZADEH, S. M. and BILLARD, A., "Learning Stable Non-Linear Dynamical Systems with Gaussian Mixture Models," *IEEE Transaction on Robotics*, 2011.

[41] KOBER, J., BAGNELL, J. A., and PETERS, J., "Reinforcement learning in robotics: A survey," *The Intl. Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[42] KORMUSHEV, P., CALINON, S., and CALDWELL, D. G., "Robot motor skill coordination with em-based reinforcement learning," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[43] KORMUSHEV, P., CALINON, S., and CALDWELL, D. G., "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[44] KRONANDER, K. and BILLARD, A., "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *Haptics, IEEE Transactions on*, vol. 7, no. 3, pp. 367–380, 2014.

[45] KULIĆ, D., OTT, C., LEE, D., ISHIKAWA, J., and NAKAMURA, Y., "Incremental learning of full body motion primitives and their sequencing through human motion observation," *The Intl. Journal of Robotics Research*, vol. 31, no. 3, pp. 330–345, 2012.

[46] KURENKOV, A., AKGUN, B., and THOMAZ, A., "An evaluation of gui and kinesthetic teaching methods for constrained-keyframe skills," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[47] LEVAS, A. and SELFRIDGE, M., "A user-friendly high-level robot teaching system," in *Proceedings of the IEEE Intl. Conference on Robotics*, (Atlanta, Georgia), pp. 413–416, 1984.

[48] LOPEZ INFANTE, M. and KYRKI, V., "Usability of force-based controllers in physical human-robot interaction," in *Proceedings of the 6th Intl. conference on Human-robot interaction*, HRI '11, pp. 355–362, 2011.

[49] LOWE, D., "Three-dimensional object recognition from single two-dimensional images," *Artificial intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[50] MELTZOFF, A. N. and DECETY, J., "What imitation tells us about social cognition: a rapprochement between developmental psychology and cognitive neuroscience," *Philosophical Transactions of the Royal Society of London*, vol. 358, pp. 491–500, 2003.

[51] MIYAMOTO, H., SCHAAL, S., GANDOLFO, F., GOMI, H., KOIKE, Y., OSU, R., NAKANO, E., WADA, Y., and KAWATO, M., "A kendama learning robot based on bi-directional theory," *Neural Netw.*, vol. 9, pp. 1281–1302, November 1996.

[52] MÜLLING, K., KOBER, J., KROEMER, O., and PETERS, J., "Learning to select and generalize striking movements in robot table tennis," *The Intl. Journal of Robotics Research*, vol. 32, no. 3, pp. 263–279, 2013.

[53] NICOLESCU, M. N. and MATARIĆ, M. J., "Natural methods for robot task learning: Instructive demonstrations, generalization and practice," in *Proceedings of the 2nd Intl. Conf. AAMAS*, (Melbourne, Australia), July 2003.

[54] NIEKUM, S., CHITTA, S., MARTHI, B., OSENTOSKI, S., and BARTO, A. G., "Incremental semantically grounded learning from demonstration," in *Robotics: Science and Systems*, vol. 9, 2013.

[55] NIEKUM, S., OSENTOSKI, S., KONIDARIS, G. D., CHITTA, S., MARTHI, B., and BARTO, A. G., "Learning grounded finite-state representations from unstructured demonstrations," *Intl. Journal of Robotics Research*, vol. 34, no. 2, pp. 131–157, 2015.

[56] PARDOWITZ, M., KNOOP, S., DILLMANN, R., and ZOLLNER, R., "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 322–332, 2007.

[57] PARENT, R., *Computer animation: algorithms and techniques*. Morgan Kaufmann series in computer graphics and geometric modeling, Morgan Kaufmann, 2002.

[58] PASTOR, P., HOFFMANN, H., ASFOUR, T., and SCHAAL, S., "Learning and generalization of motor skills by learning from demonstration," in *IEEE Intl. Conference on Robotics and Automation*, 2009.

[59] PASTOR, P., KALAKRISHNAN, M., CHITTA, S., THEODOROU, E., and SCHAAL, S., "Skill learning and task outcome prediction for manipulation," in *2011 IEEE Intl. Conference on Robotics and Automation (ICRA)*, 2011.

[60] PASTOR, P., KALAKRISHNAN, M., MEIER, F., STULP, F., BUCHLI, J., THEODOROU, E., and SCHAAL, S., "From dynamic movement primitives to associative skill memories," *Robotics and Autonomous Systems*, 2012.

[61] RATLIFF, N., ZIEBART, B., PETERSON, K., BAGNELL, J. A., HEBERT, M., DEY, A. K., and SRINIVASA, S., "Inverse optimal heuristic control for imitation learning," in *Proc. AISTATS*, pp. 424–431, 2009.

[62] ROZO, L., JIMENEZ, P., and TORRAS, C., "Robot learning from demonstration of force-based tasks with multiple solution trajectories," in *Advanced Robotics (ICAR), 2011 15th International Conference on*, pp. 124–129, IEEE, 2011.

[63] RUSU, R. B., BRADSKI, G., THIBAUX, R., and HSU, J., "Fast 3d recognition and pose using the viewpoint feature histogram," in *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS)*, 2010.

[64] SCHULMAN, J., HO, J., LEE, C., and ABBEEL, P., "Learning from demonstrations through the use of non-rigid registration," in *Proceedings of the 16th International Symposium on Robotics Research (ISRR)*, 2013.

[65] SILVER, D., BAGNELL, J. A., and STENTZ, A., "Learning from demonstration for autonomous navigation in complex unstructured terrain," *The International Journal of Robotics Research*, 2010.

[66] STARNER, T. and PENTLAND, A., "Real-time american sign language recognition from video using hidden markov models," in *Motion-Based Recognition*, pp. 227–243, Springer, 1997.

[67] SUAY, H. B., TORIS, R., and CHERNOVA, S., "A practical comparison of three robot learning from demonstration algorithms," *Intl. Journal of Social Robotics, Special Issue on Robot Learning from Demonstration*, 2012.

[68] THOMAZ, A. L. and BREAZEAL, C., "Experiments in socially guided exploration: Lessons learned in building robots that learn with and without human teachers.," *Connection Science, Special Issue on Social Learning in Embodied Agents, forthcoming*, 2008.

[69] THOMAZ, A. L. and BREAZEAL, C., "Teachable robots: Understanding human teaching behavior to build more effective robot learners.," *Artificial Intelligence Journal*, vol. 172, pp. 716–737, 2008.

[70] TODOROV, E. and JORDAN, M., "Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements," *Journal of Neurophysiology*, vol. 80, no. 2, pp. 696–714, 1998.

[71] TREVOR, A. J. B., GEDIKLI, S., RUSU, R. B., and CHRISTENSEN, H. I., "Efficient organized point cloud segmentation with connected components," in *Workshop on Semantic Perception, Mapping and Exploration*, 2013.

[72] WADA, Y. and KAWATO, M., "A neural network model for arm trajectory formation using forward inverse dynamics models.," *Neural Networks 6*, pp. 919–932, 1993.

[73] WEISS, A., IGELSBOECK, J., CALINON, S., BILLARD, A., and TSCHELIGI, M., "Teaching a humanoid: A user study on learning by demonstration with hoap-3," in *Proceedings of the IEEE Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 147–152, 2009.

[74] ZANG, P., TIAN, R., THOMAZ, A. L., and ISBELL, C., "Batch versus interactive learning by demonstration," in *Proceedings of the Intl. Conference on Development and Learning (ICDL 2010)*, 2010.